

FIG. 1

09437590 110999

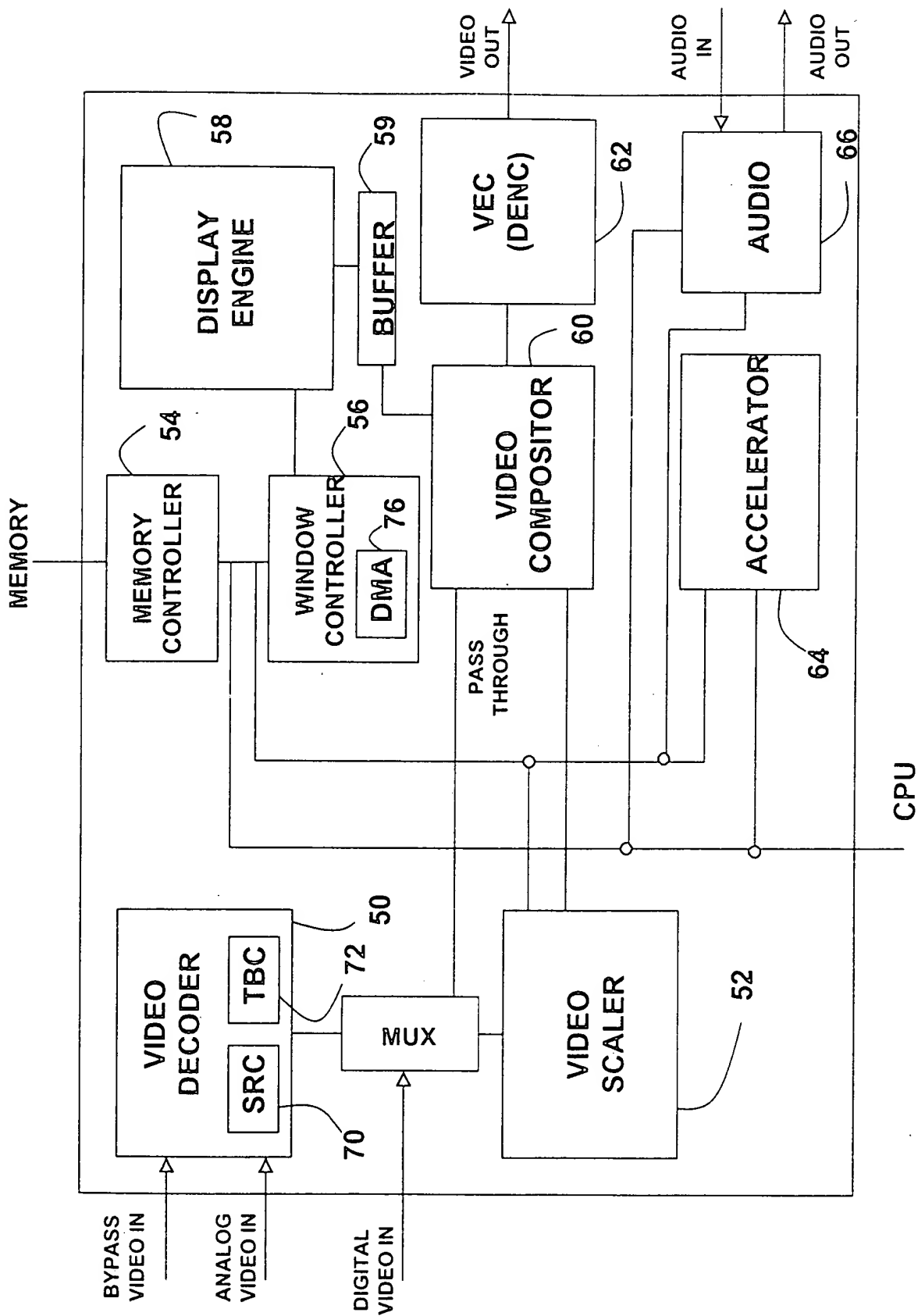


FIG. 2

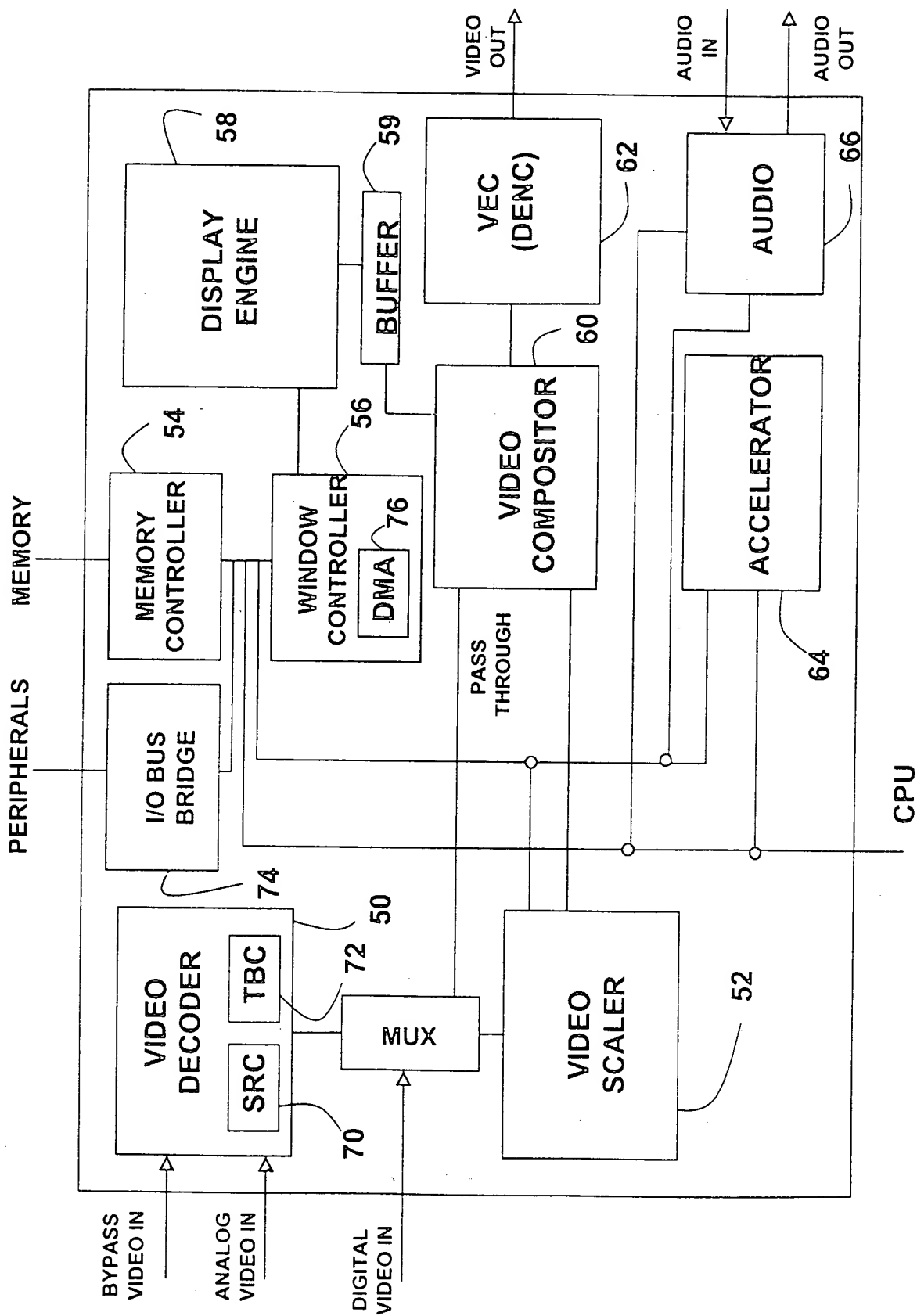


FIG. 3

09427530 110992

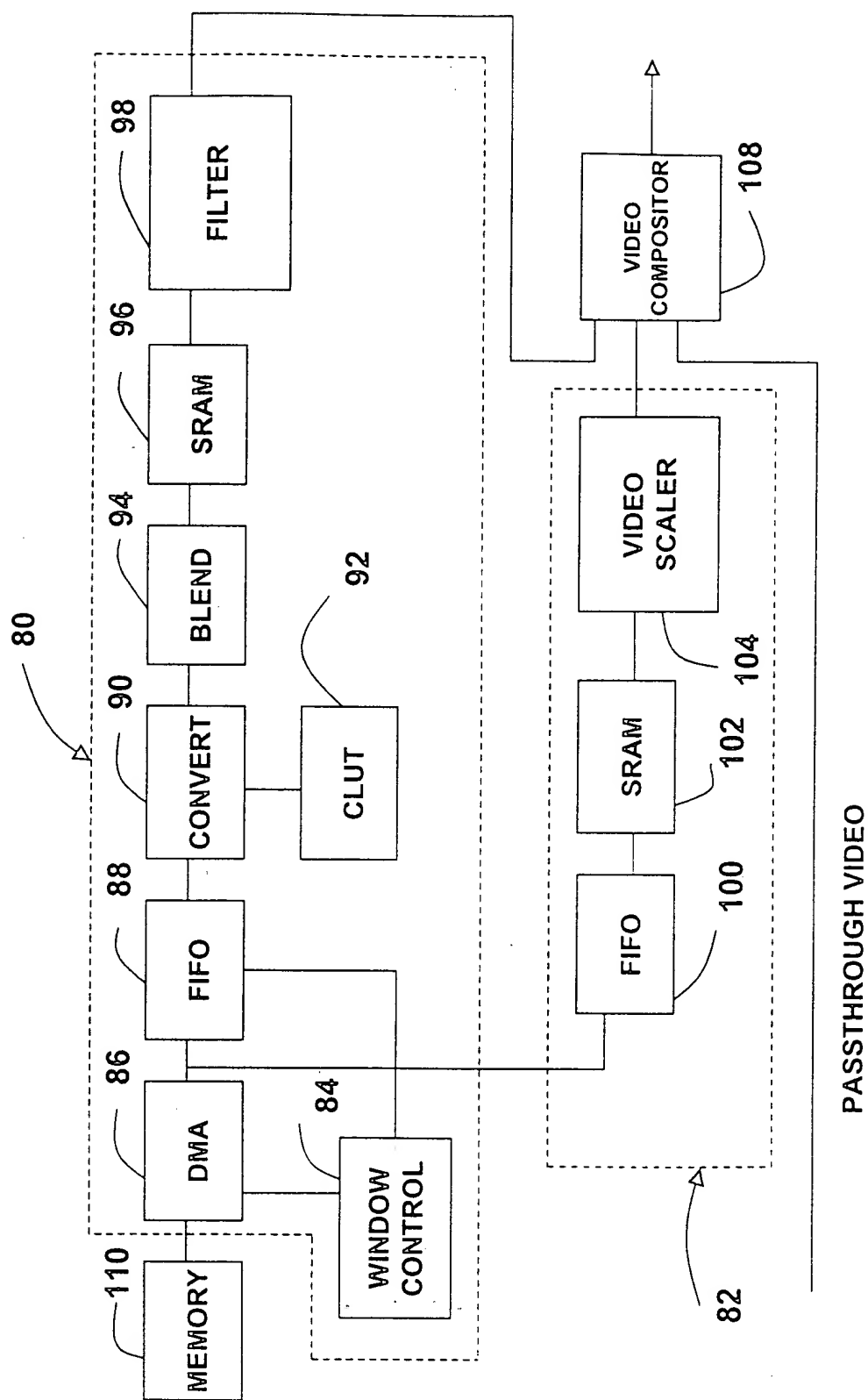


FIG. 4

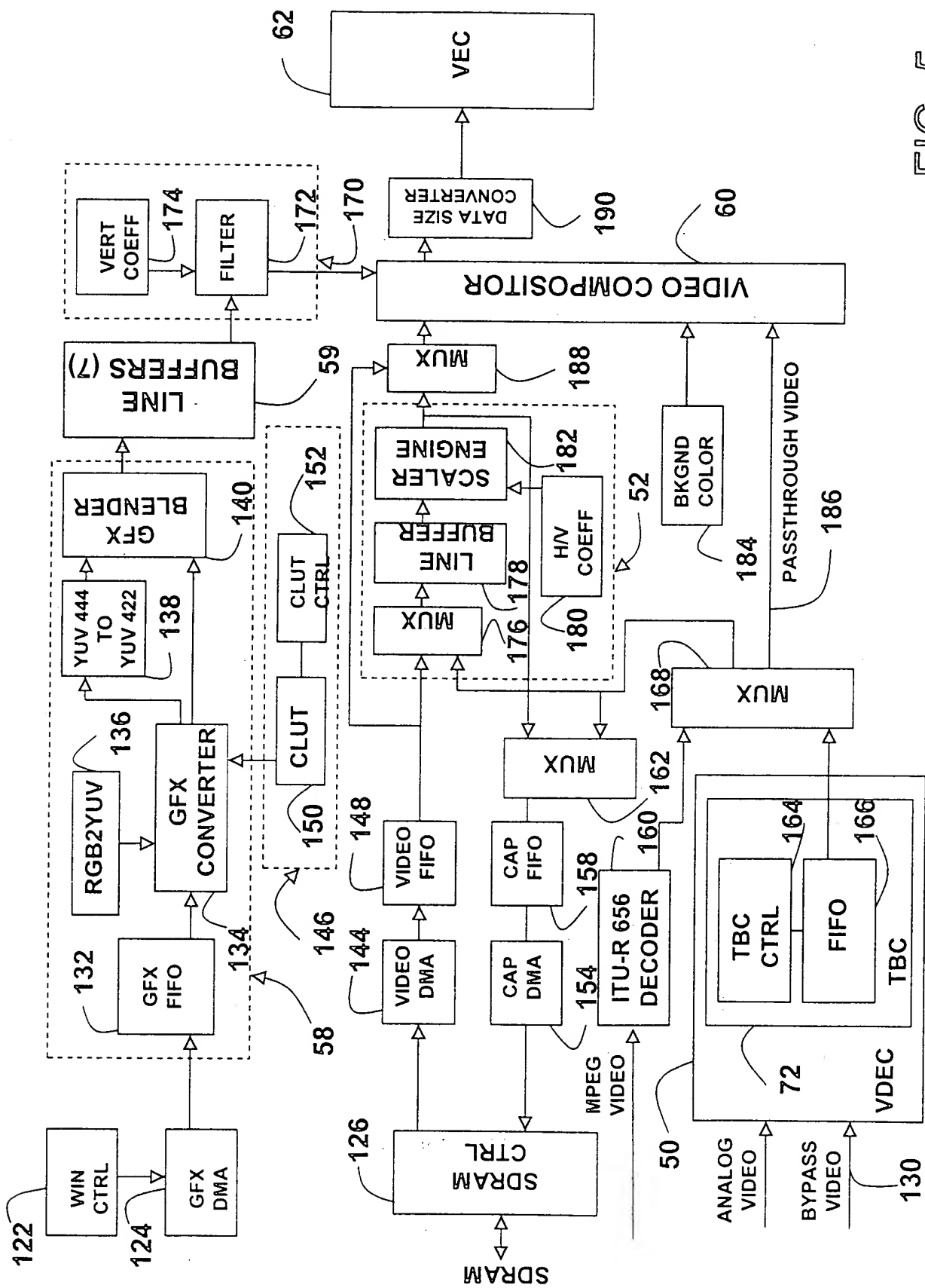


FIG. 5

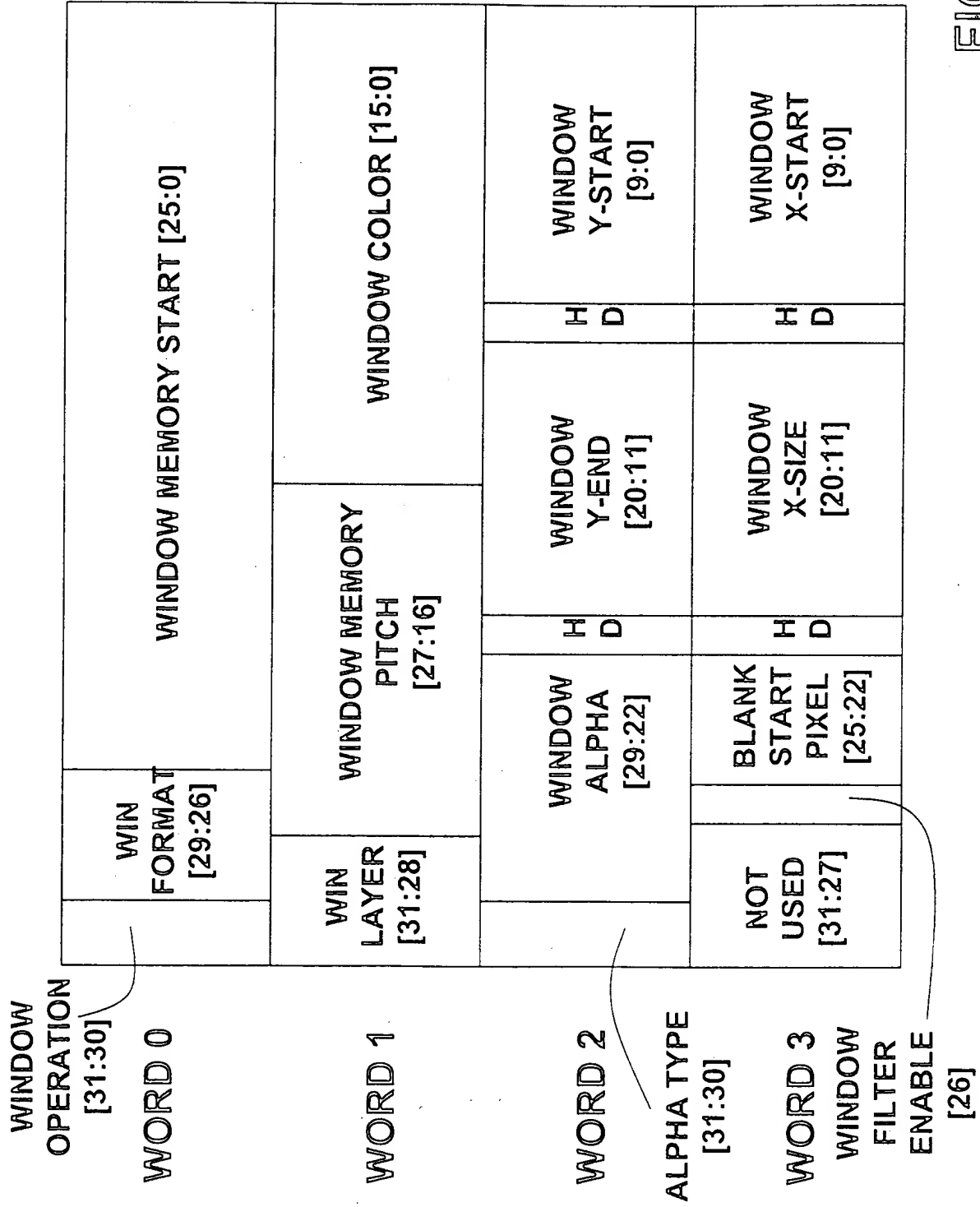


FIG. 6

09437520 110909

The diagram illustrates the system architecture for processing window descriptors. At the top, eight window descriptors (WD0 to WD7) are shown, each associated with a hexadecimal address (300a to 300h). These descriptors provide 'WD parameters' to a 'MUX' (Multiplexer) block. A 'Sorting' block (304) also receives inputs from the WD parameters and outputs to the MUX. The MUX selects the 'smallest WD parameters' (302). These parameters are then processed by a 'DMA (assemble header and request memory data)' block (306), which interacts with a 'memory controller'. The DMA block outputs 'WD header, GFX data' to a 'GFX FIFO' block (308). The output of the FIFO is 'WD header, GFX data', which is then sent to the 'GFX DISP' (Graphics Display) block.

FIG. 7

6600-0000-0000-0000

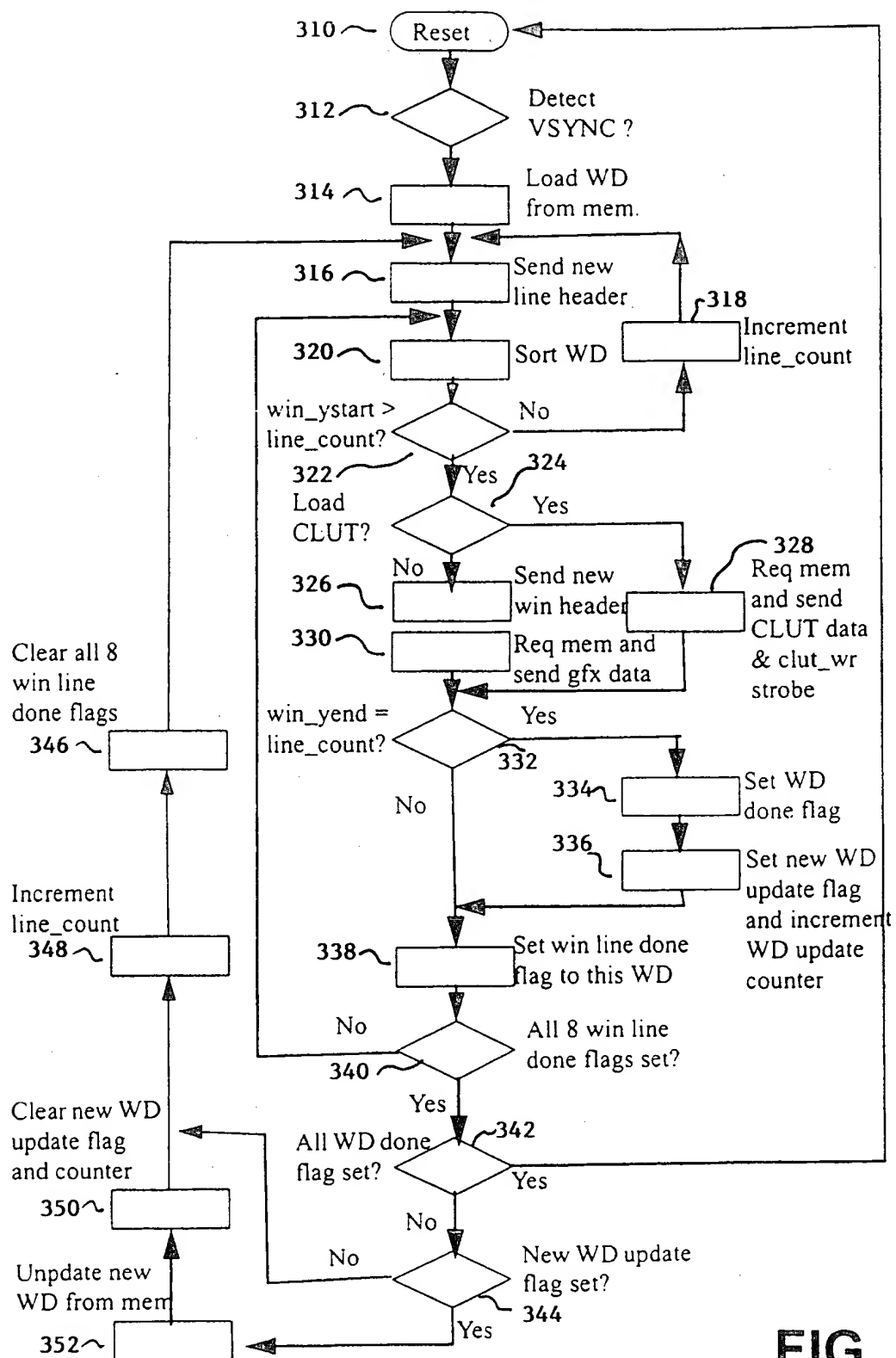


FIG. 8


```

graph TD
    354([GFX_HEADER 1]) --> 356{FIRST WINDOW ?}
    356 -- YES --> 358([CLOCK SWITCH])
    356 -- NO --> 362([GFX_HEADER 2])
    358 --> 360{GFX_TYPE = 1111 ?}
    360 -- YES --> 354
    360 -- NO --> 362
    362 --> 364([GFX_CONTENT])
    364 --> 366{WORD IS DATA ?}
    366 -- YES --> 364
    366 -- NO --> 368([PIPE_COMP])
    368 --> 354

```

FIG. 9

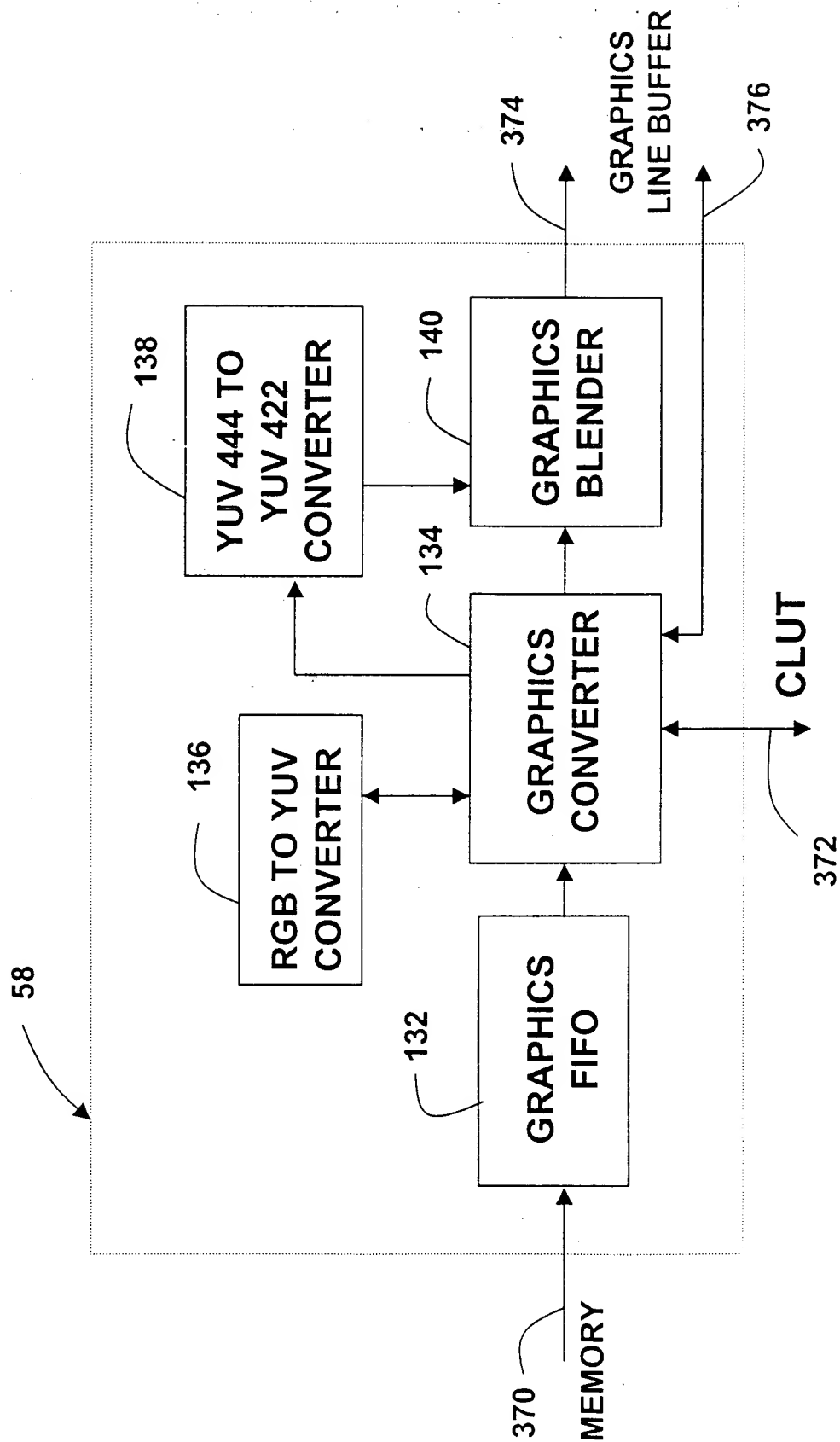


FIG. 10

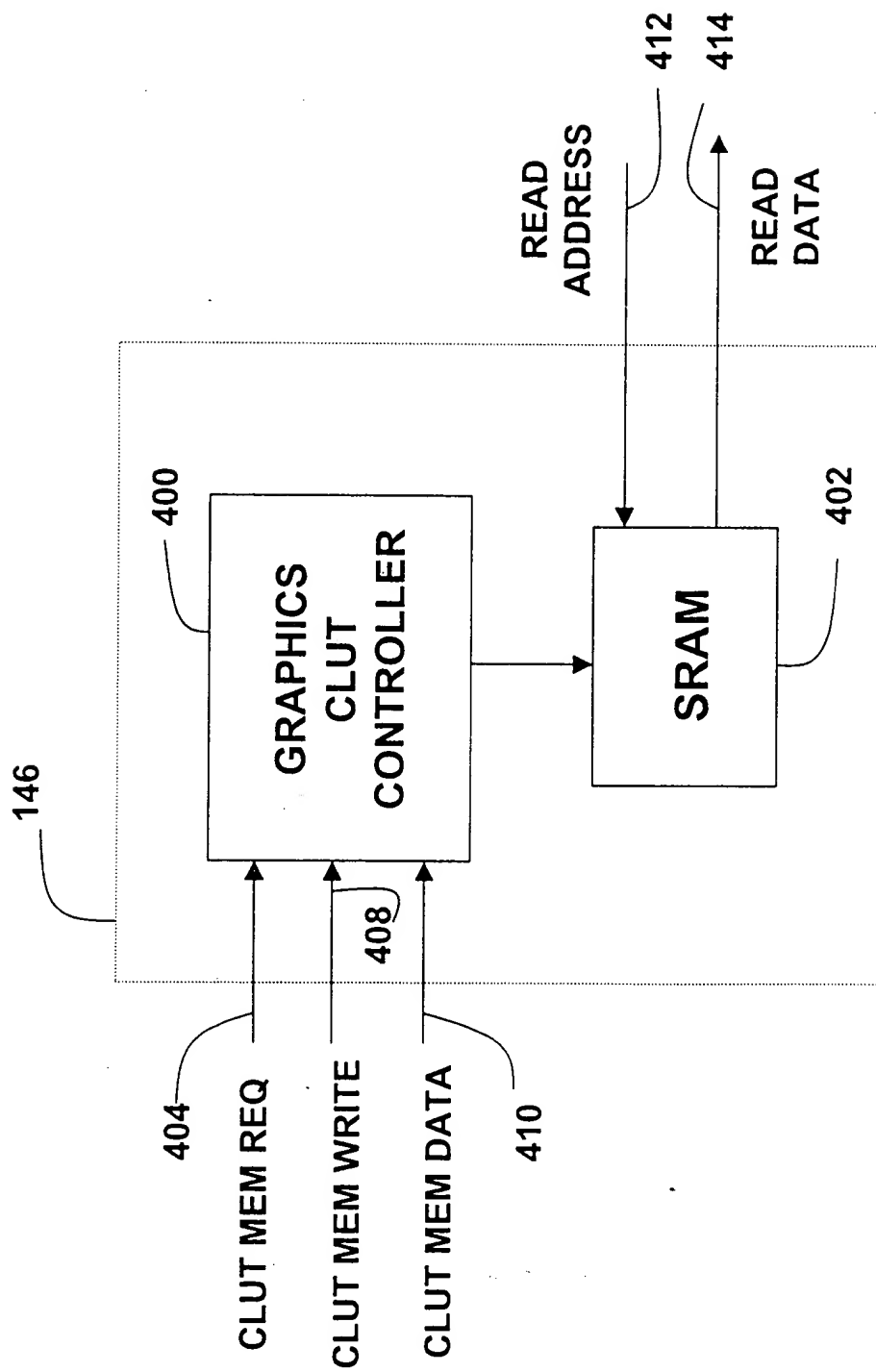
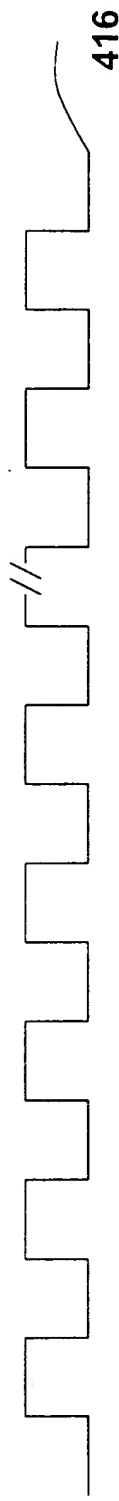
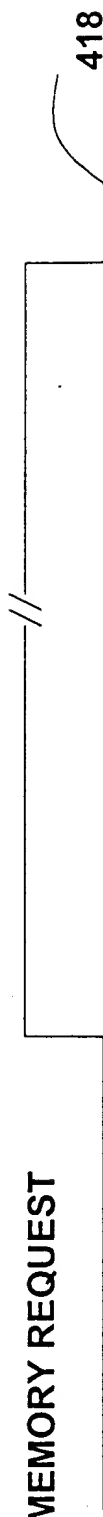


FIG. 11

MEMORY CLOCK



CLUT MEMORY REQUEST



CLUT MEMORY WRITE



CLUT MEMORY DATA



FIG. 12

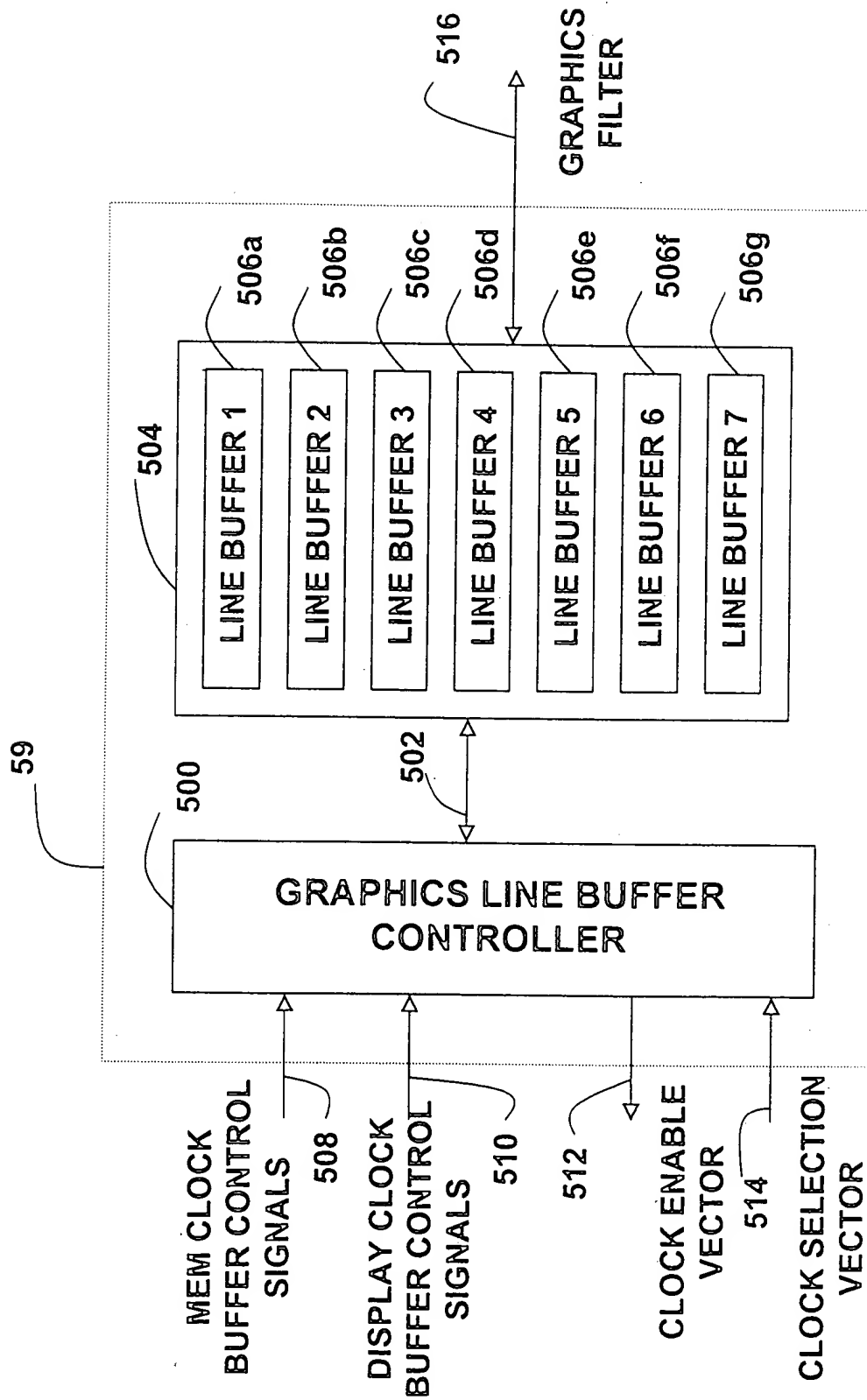


FIG. 13

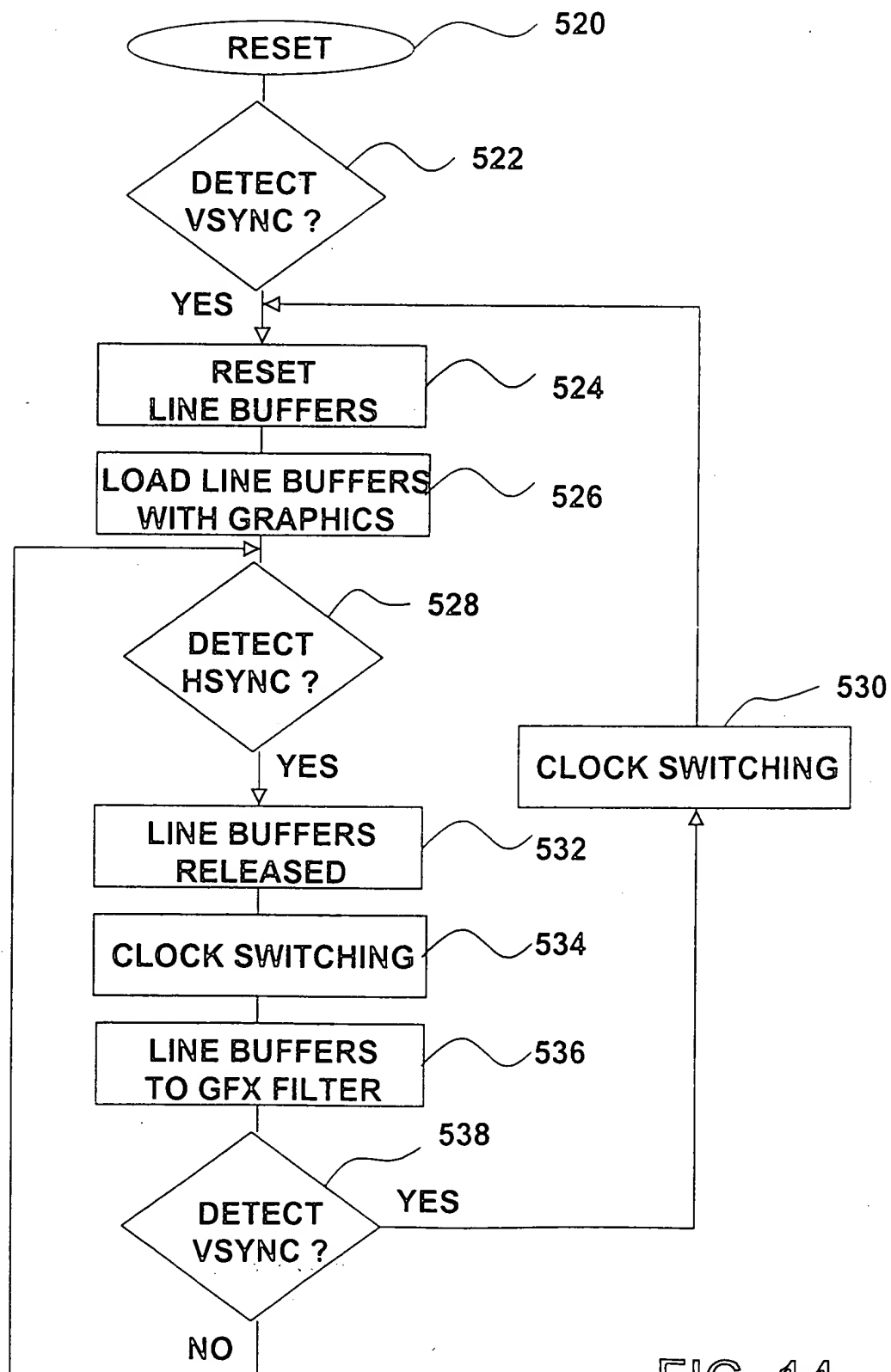


FIG. 14

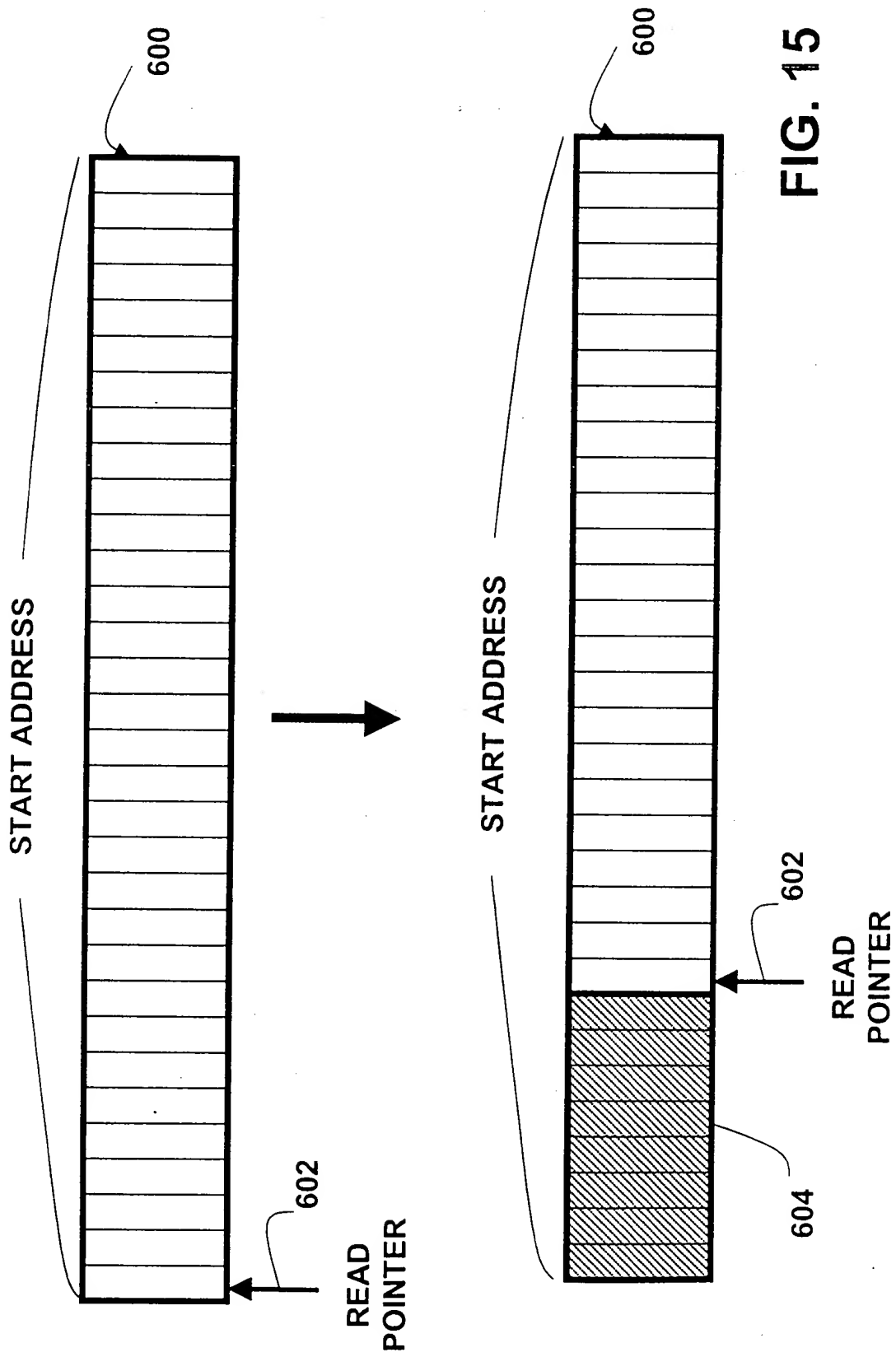


FIG. 15

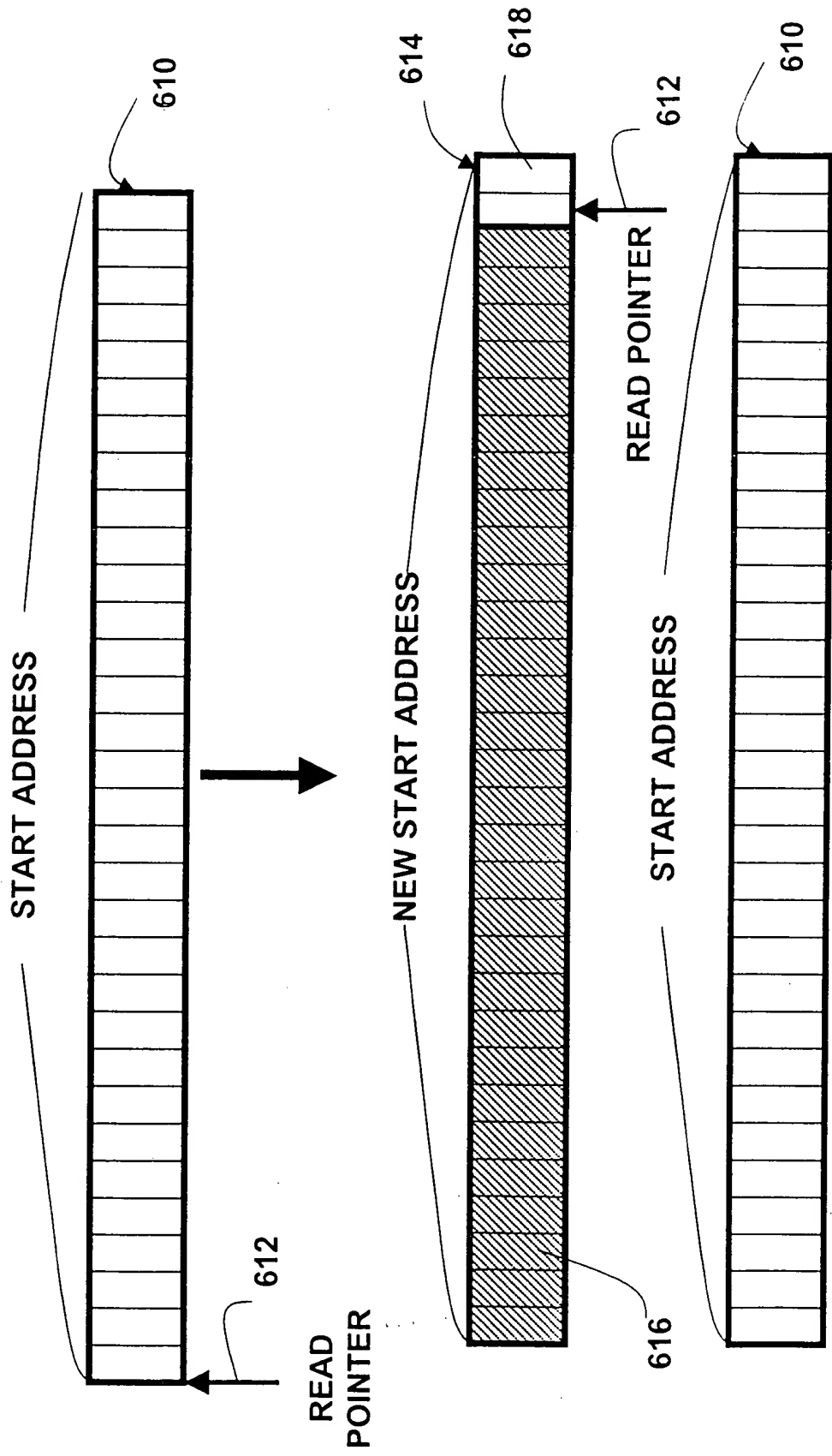


FIG. 16

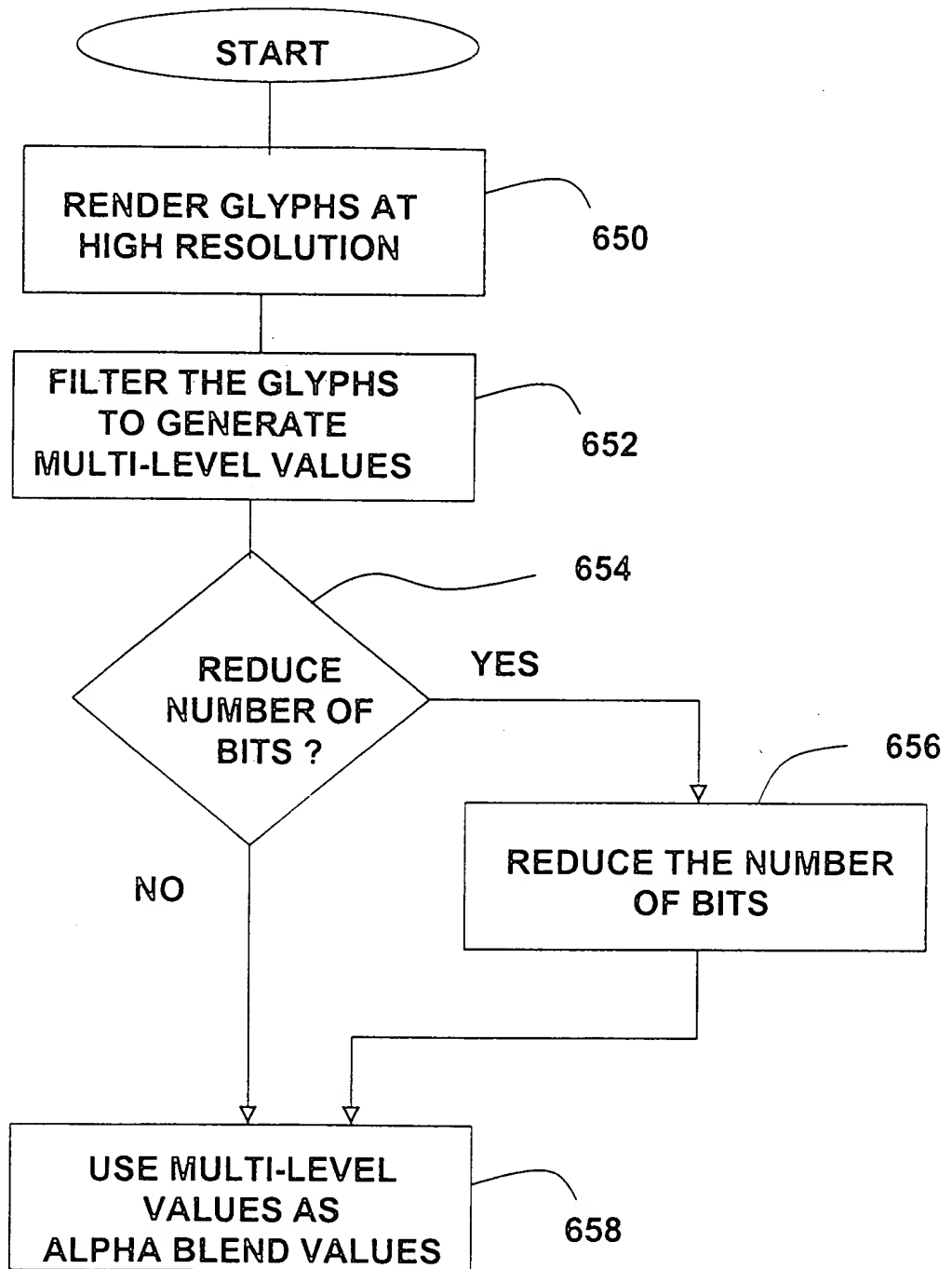


FIG. 17

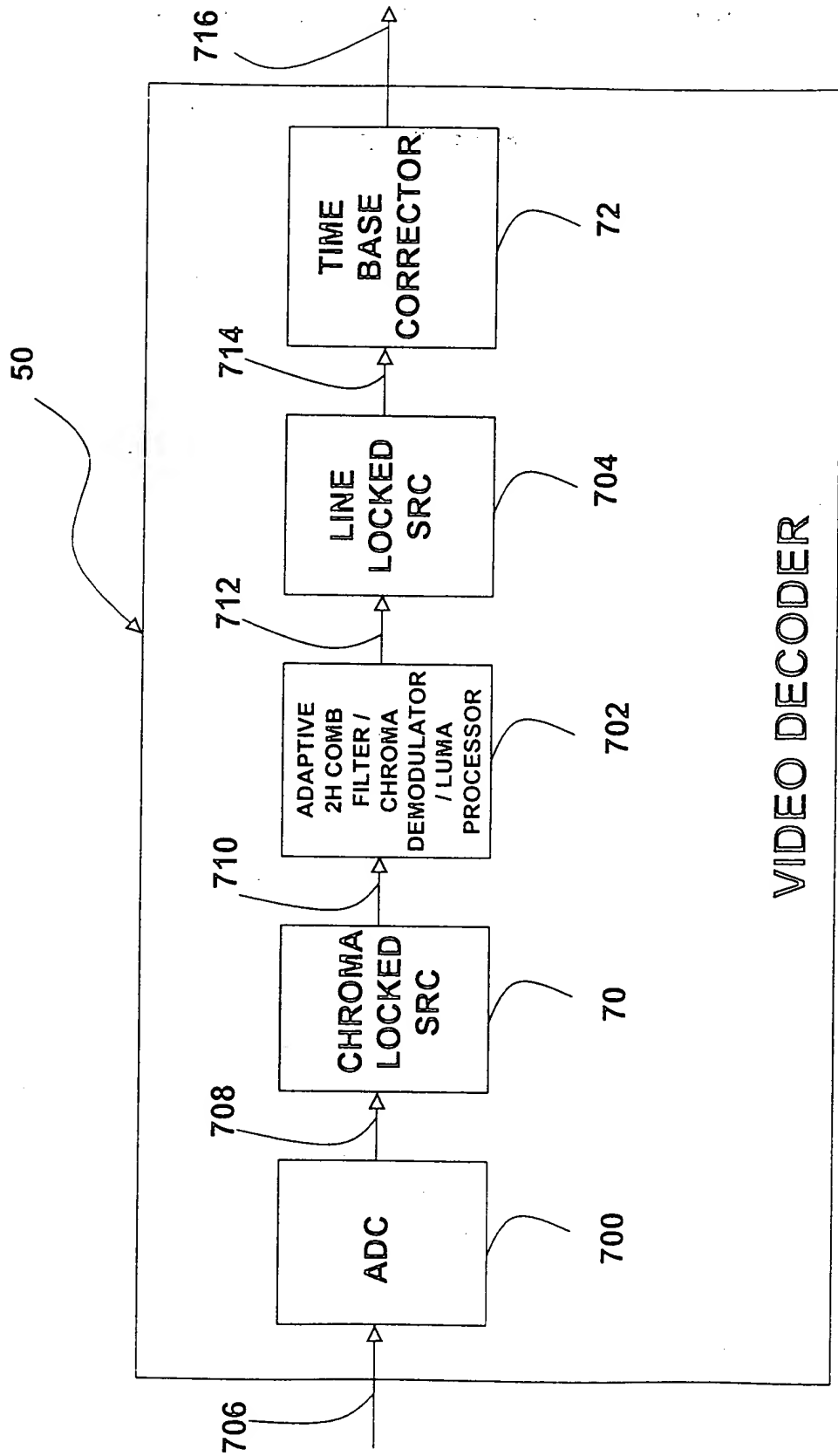


FIG. 18

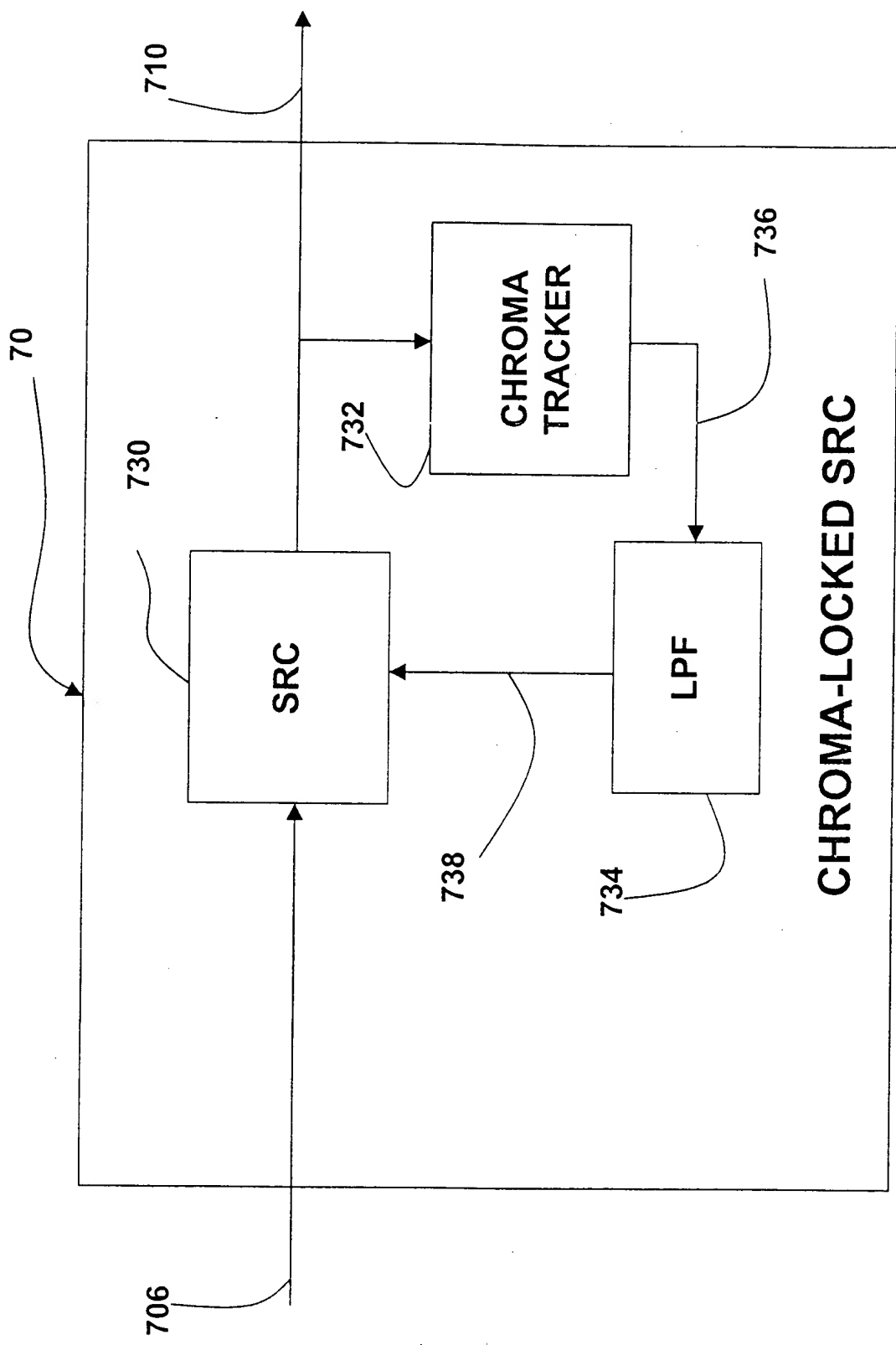


FIG. 19

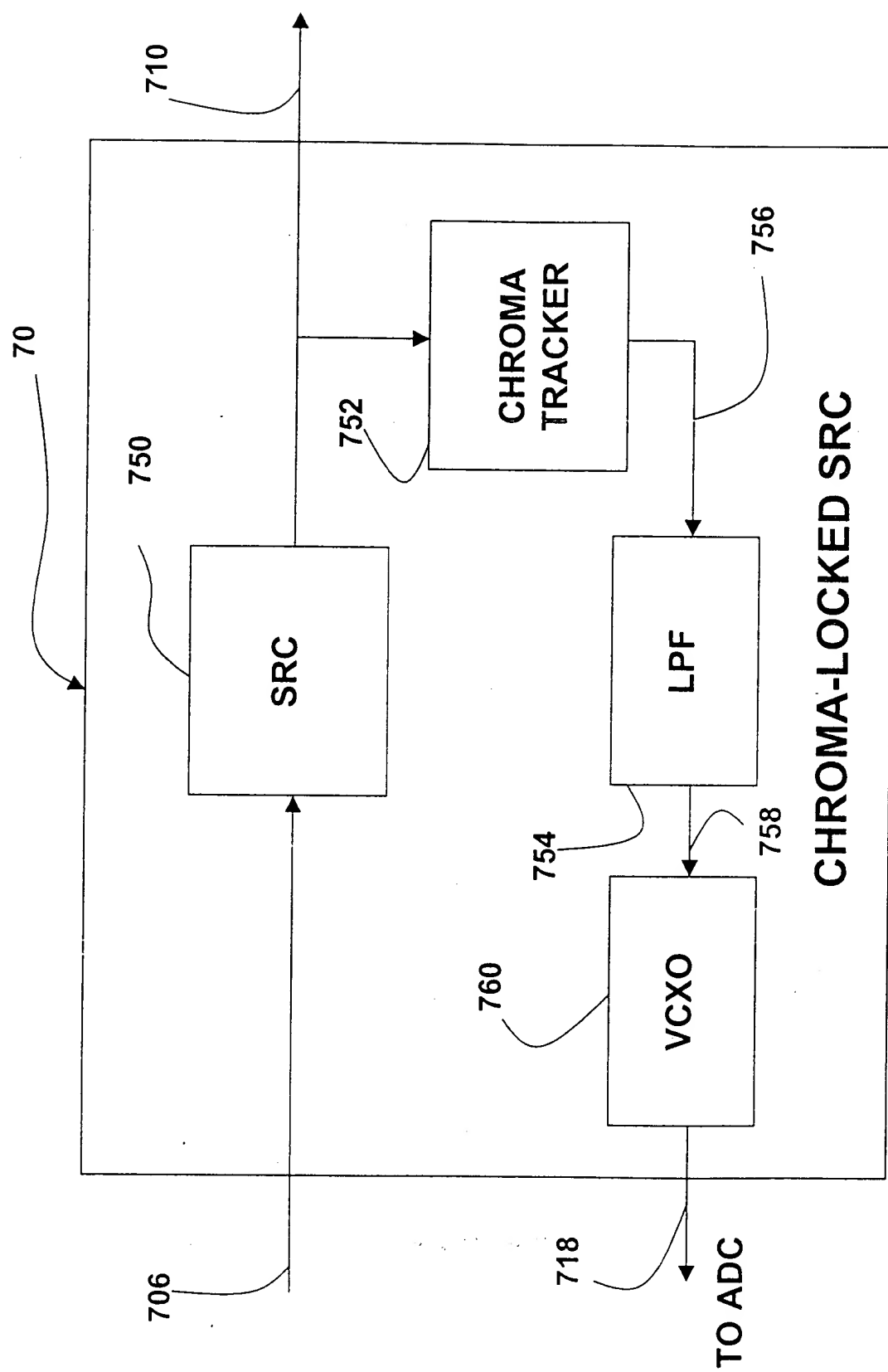


FIG. 20

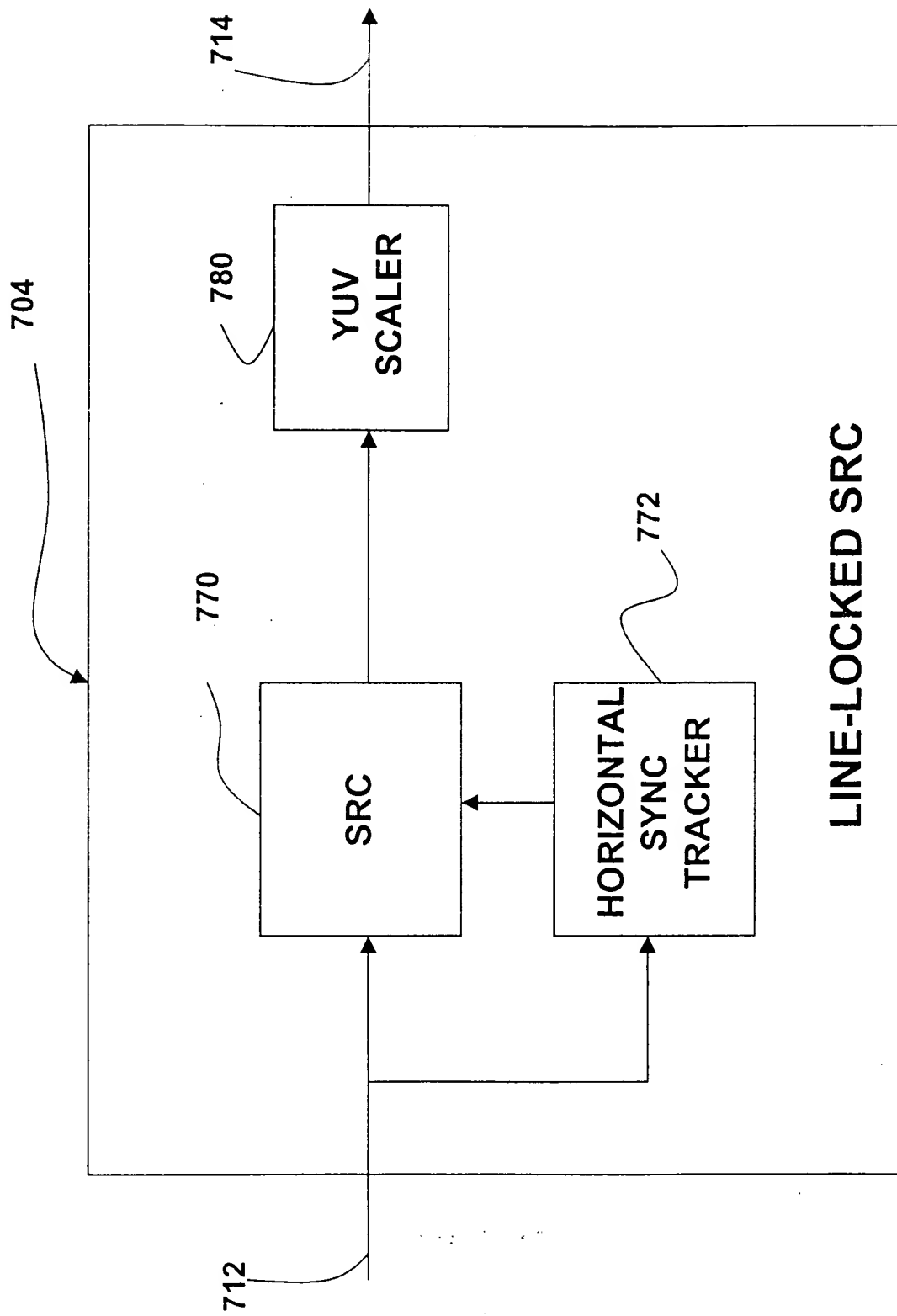


FIG. 21

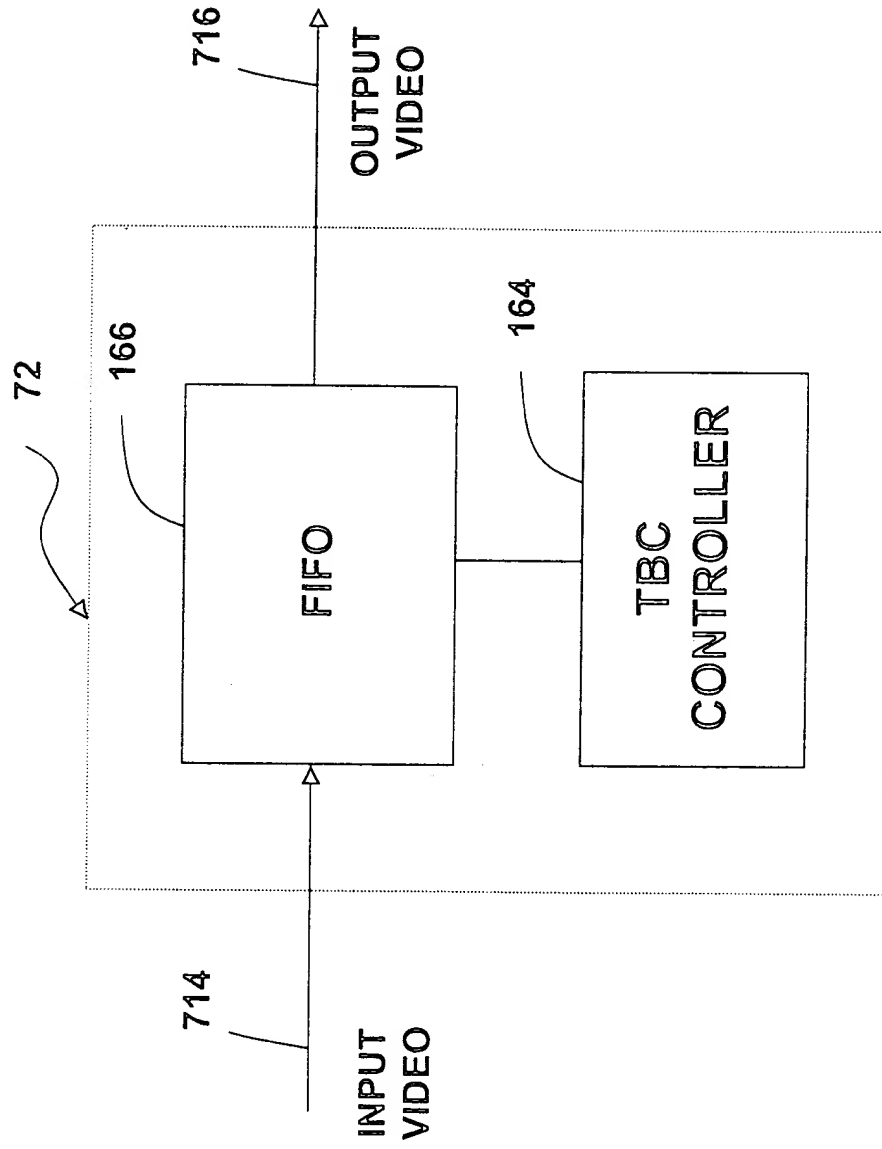


FIG. 22

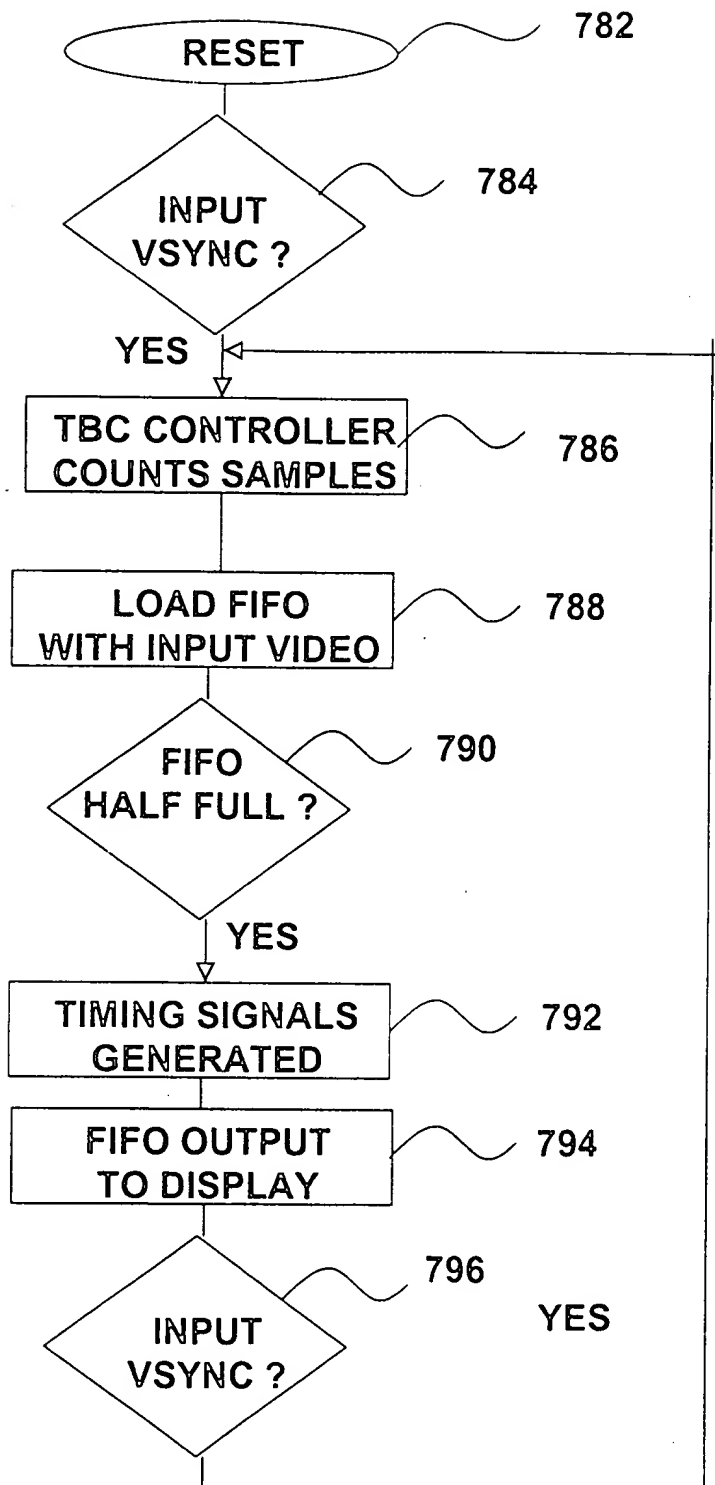


FIG. 23

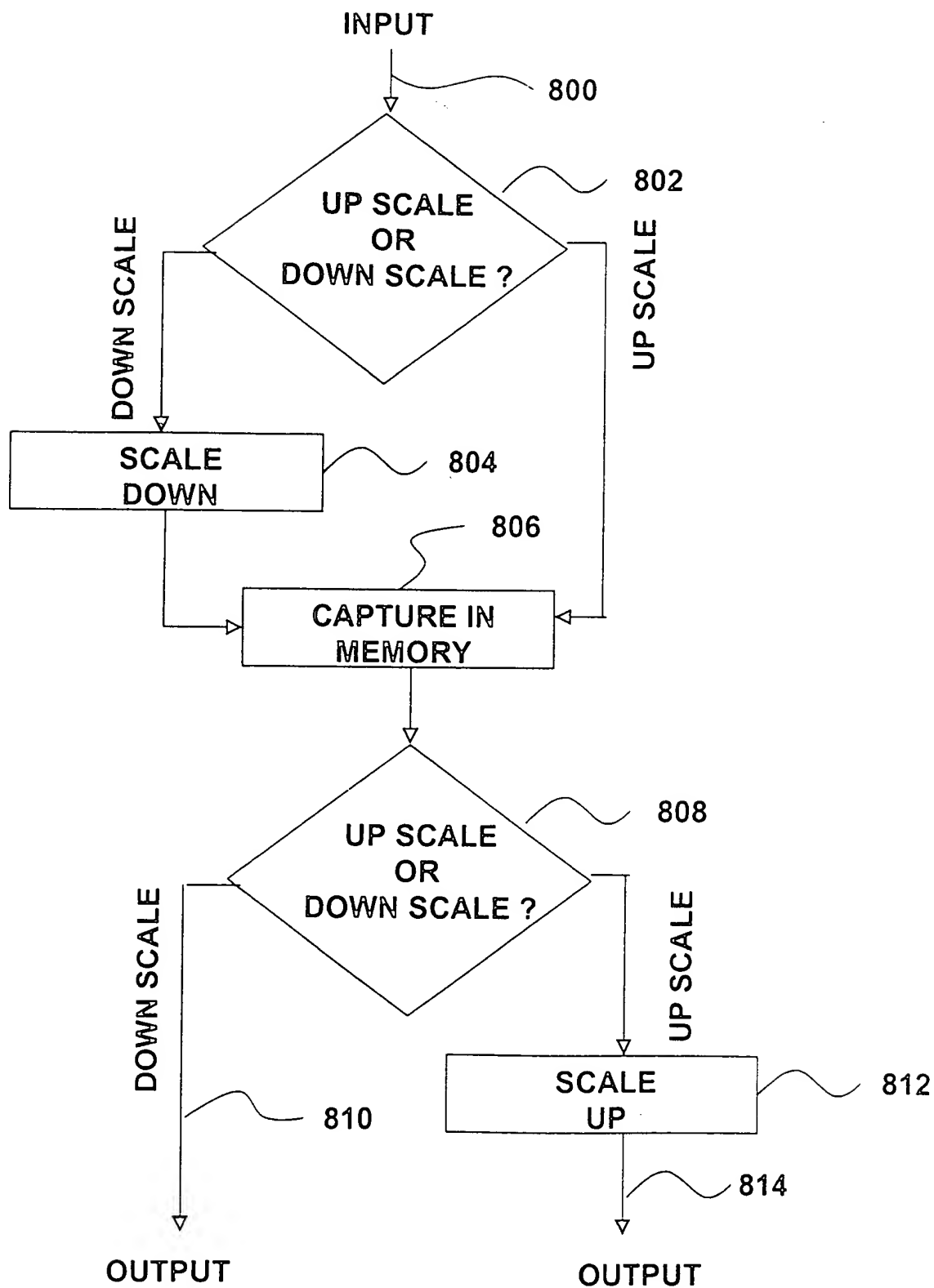


FIG. 24

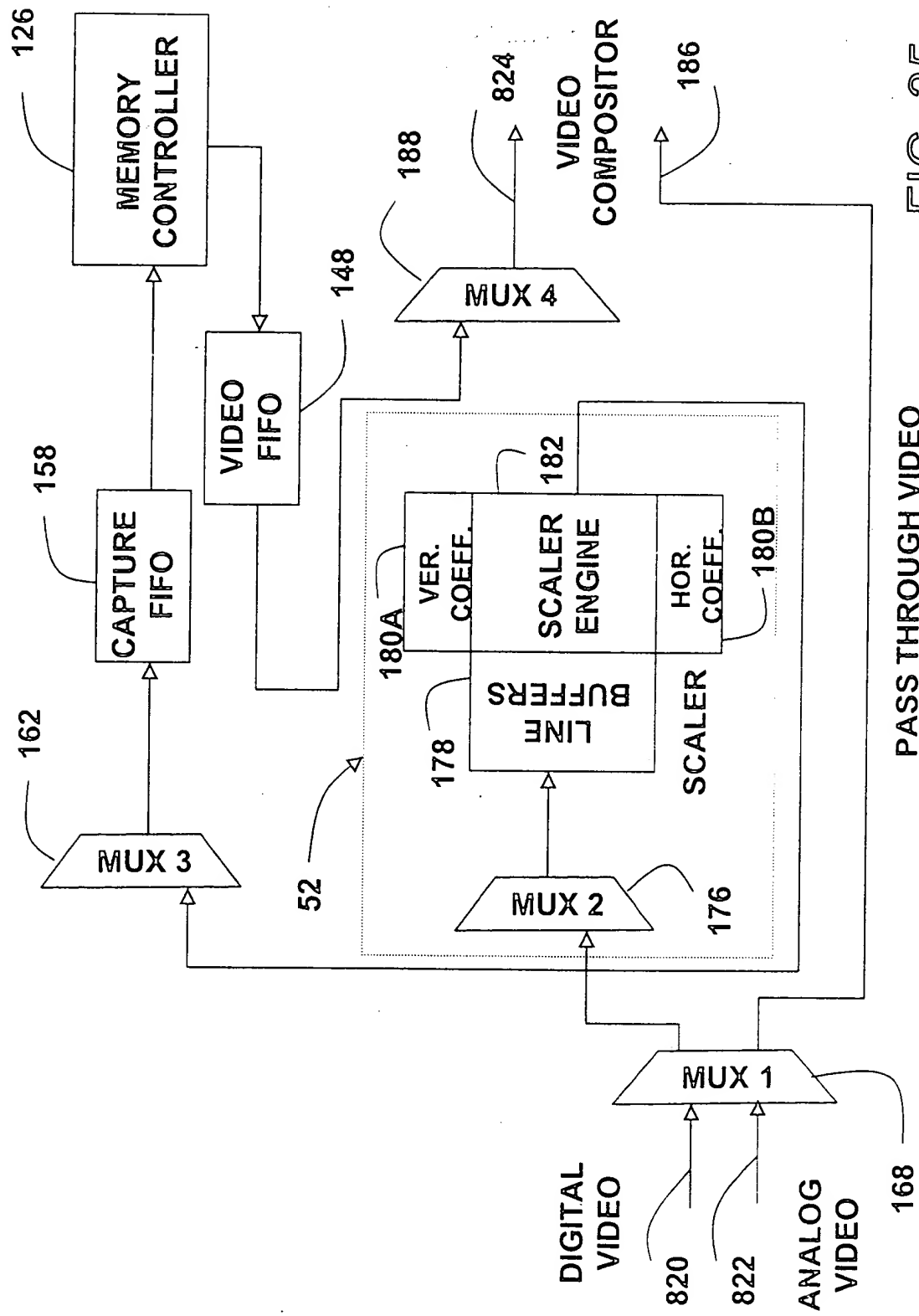


FIG. 25

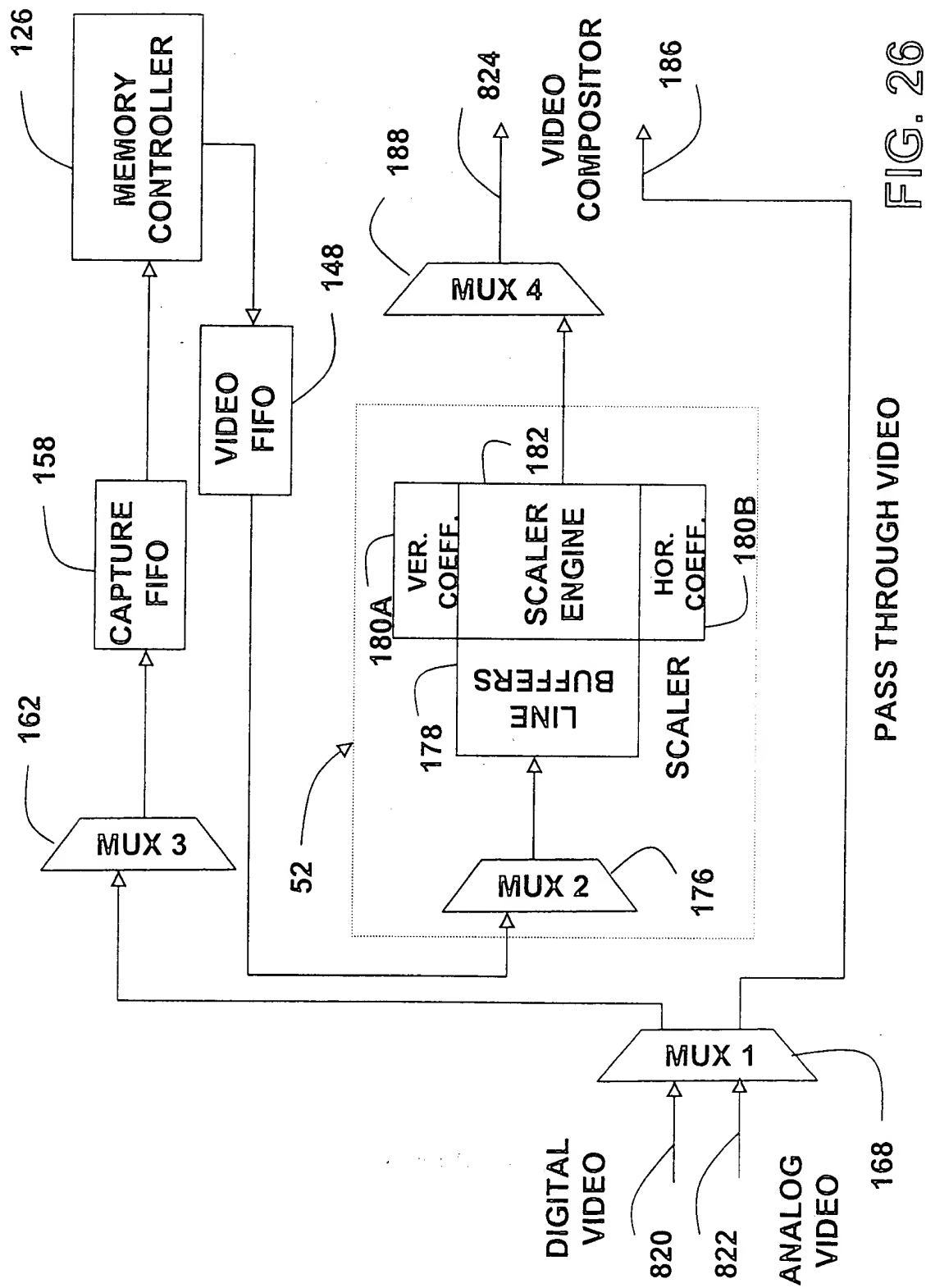


FIG. 26

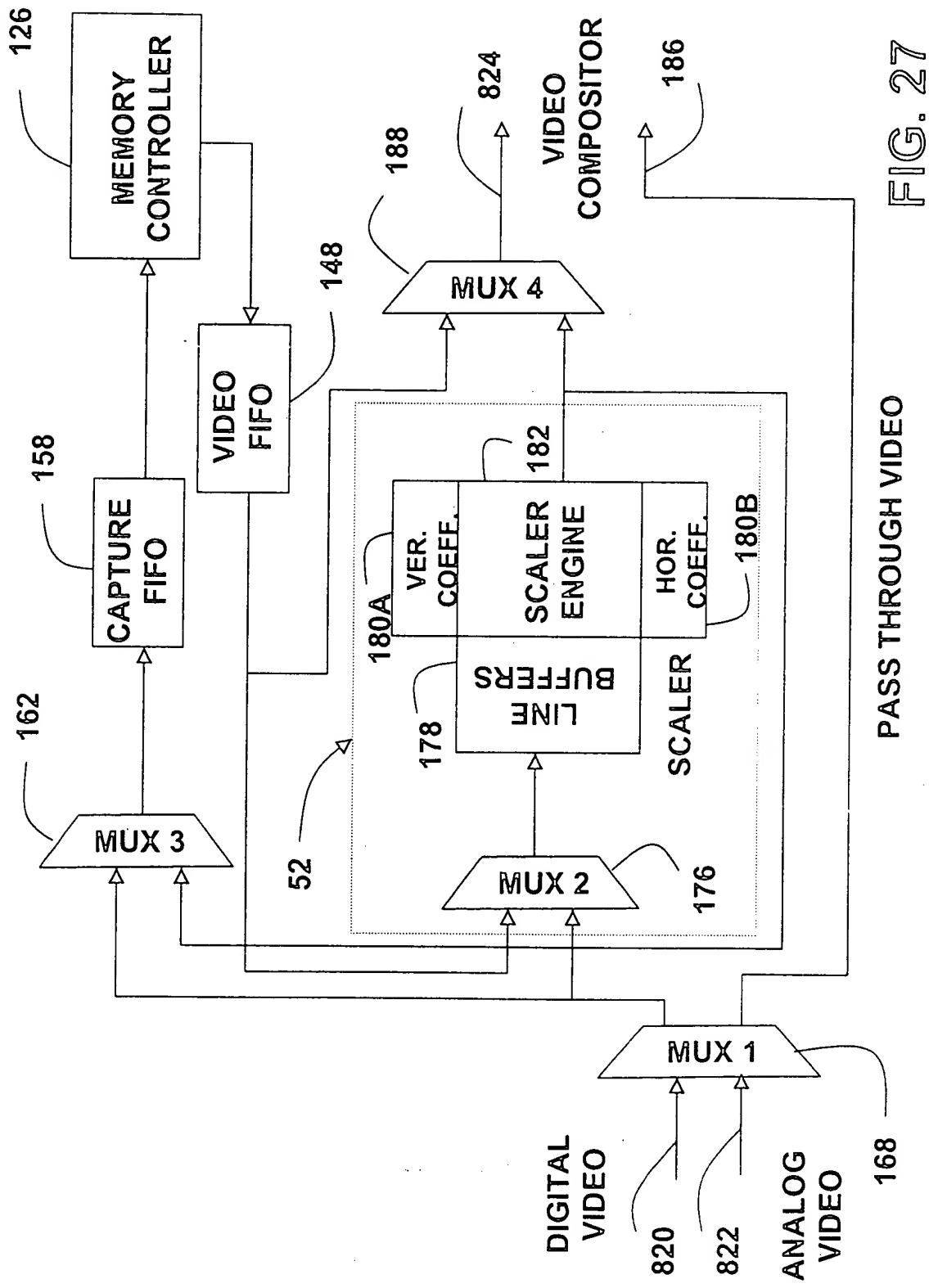


FIG. 27

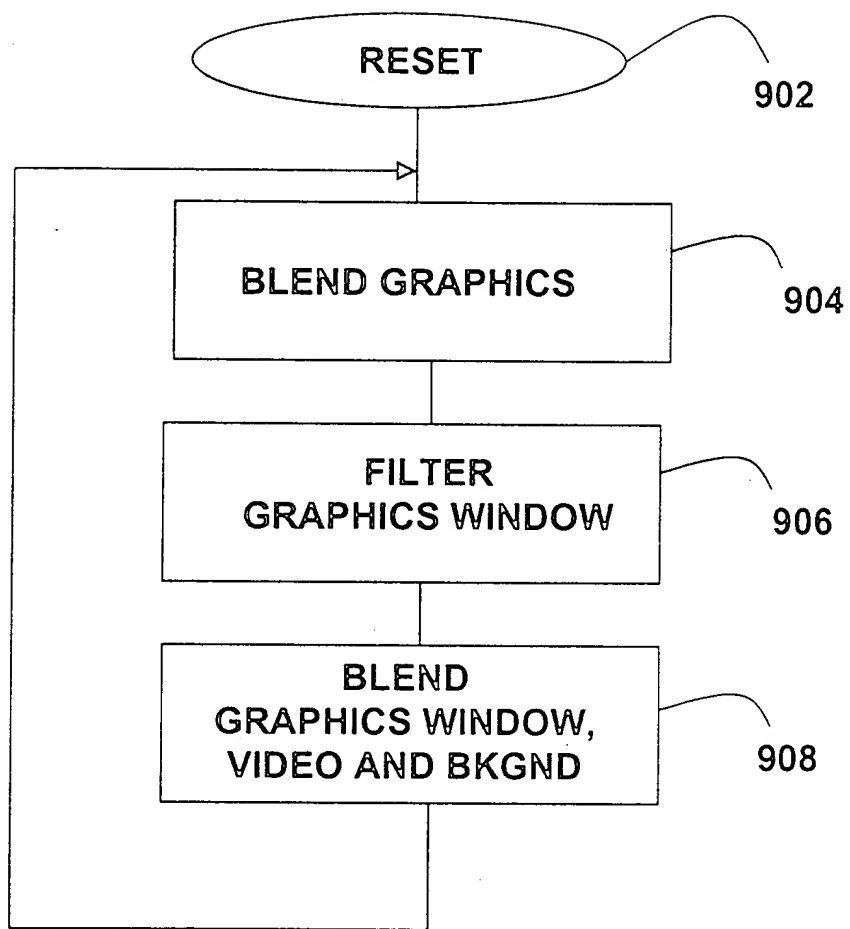


FIG. 28

```

graph TD
    920([RESET]) --> 922{VSYNC ?}
    922 --> 924[LOAD BOTTOM MOST WINDOW]
    924 --> 926[BLEND NEXT WINDOW]
    926 --> 928{LAST WINDOW ON LINE}
    928 -- NO --> 922
    928 -- YES --> 930{LAST LINE OF FIELD ?}
    930 -- YES --> 922
    930 -- NO --> 932[GO TO NEXT LINE]
    932 --> 920

```

FIG. 22

FIG. 29

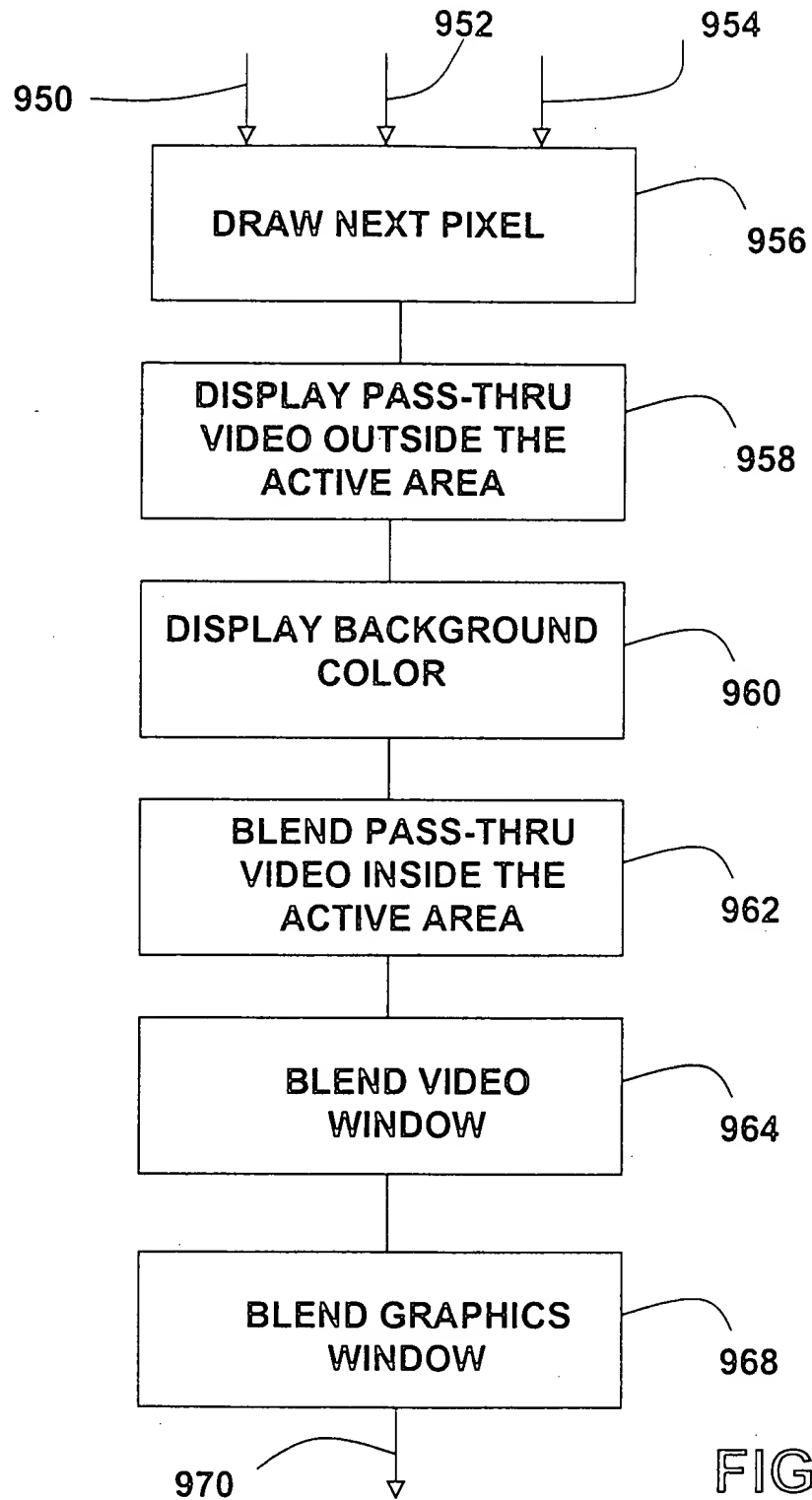


FIG. 30

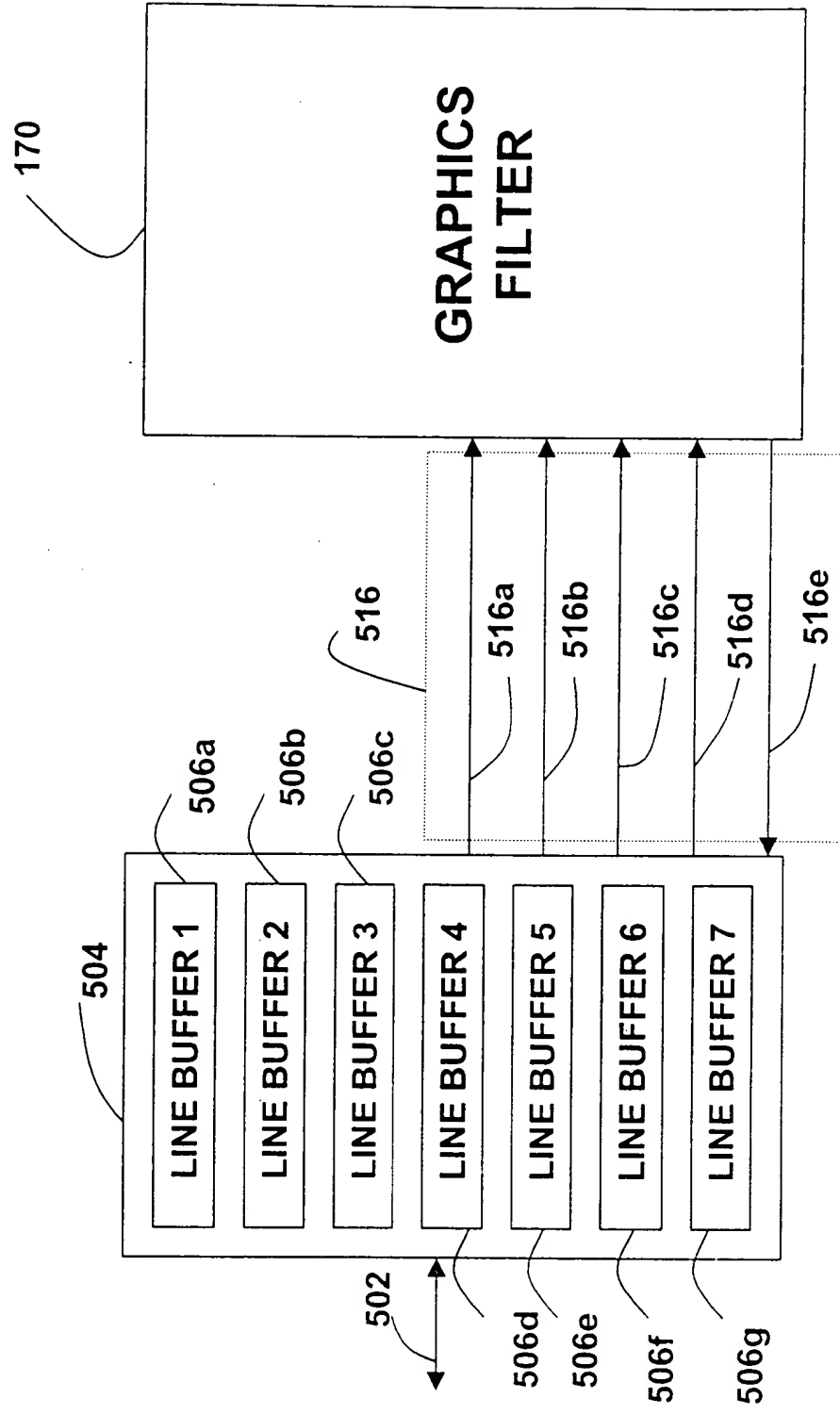


FIG. 31

Figure 1 is a block diagram of a memory access system. A central vertical block labeled "mem select" (1100) is connected to four input multiplexers (1110, 1112, 1114, 1116) and two output arbiters (1102, 1106). The input multiplexers select between GFX (1118), CPU (1120), GFX ACCEL. (1124), and IO (1126). The output arbiters select between mem_ctrl 0 (1104) and mem_ctrl 1 (1108). The system is shown with multiple data paths and control signals.

[illegible]

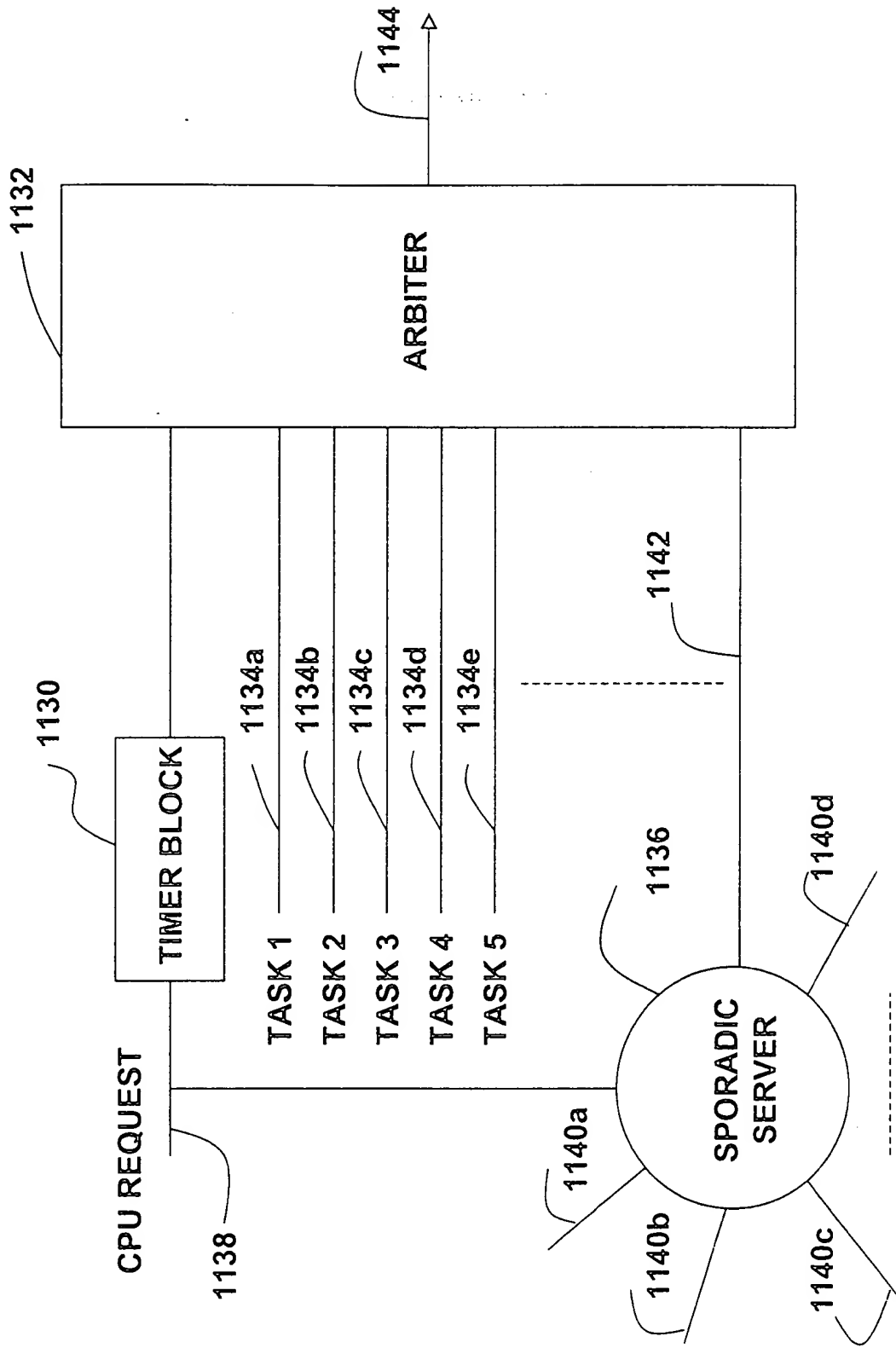


FIG. 33

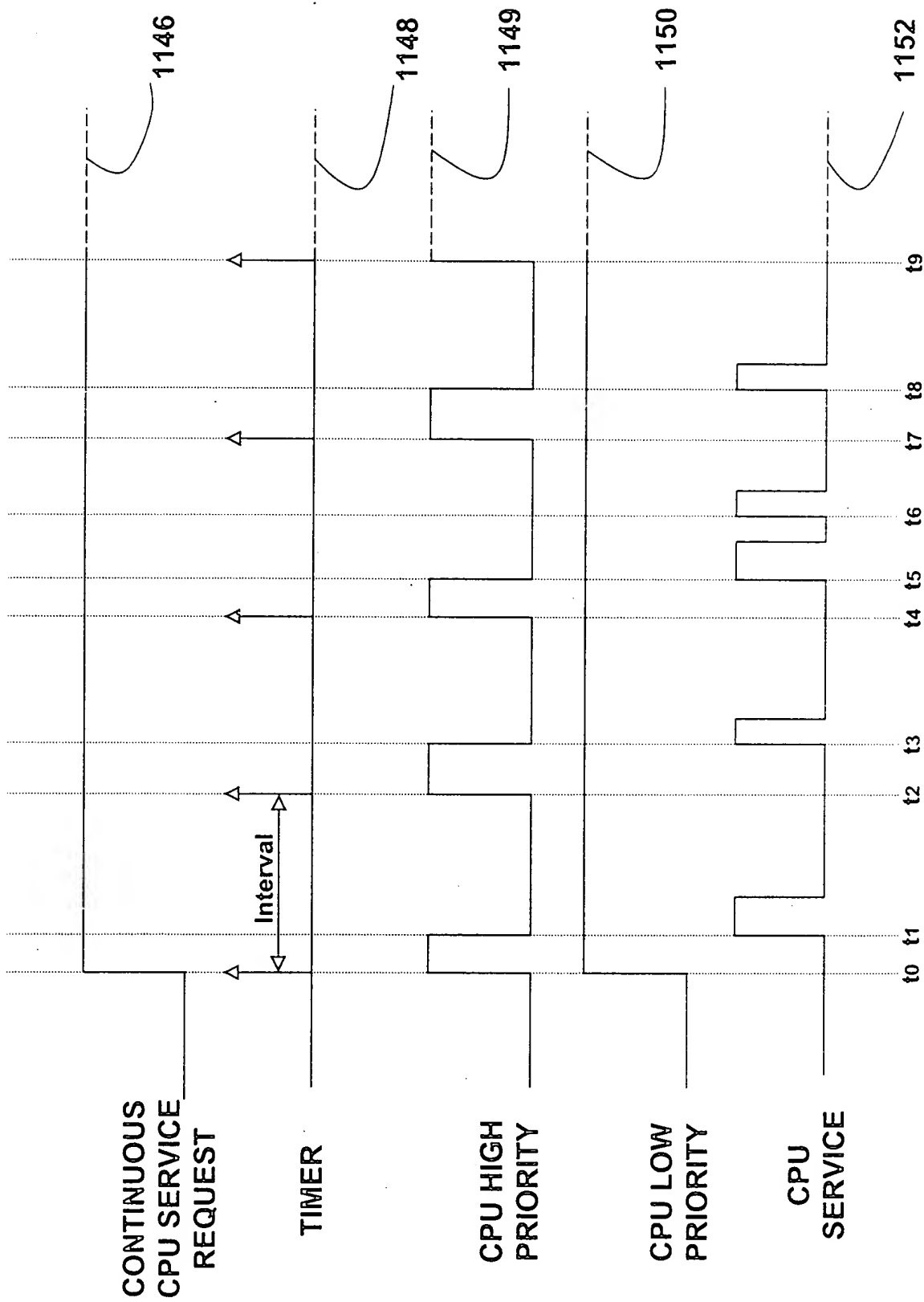


FIG. 34

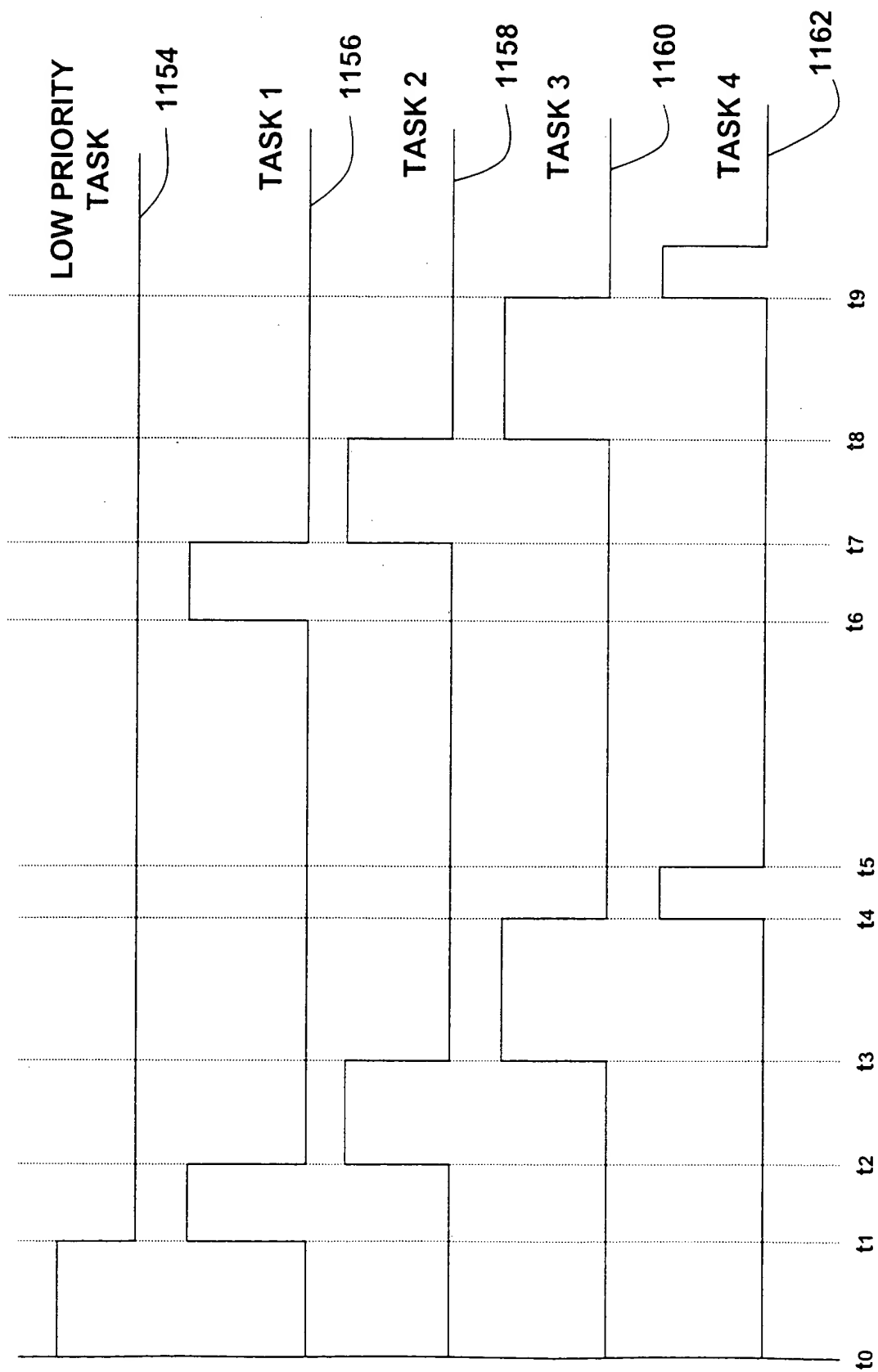


FIG. 35

Highest Priority

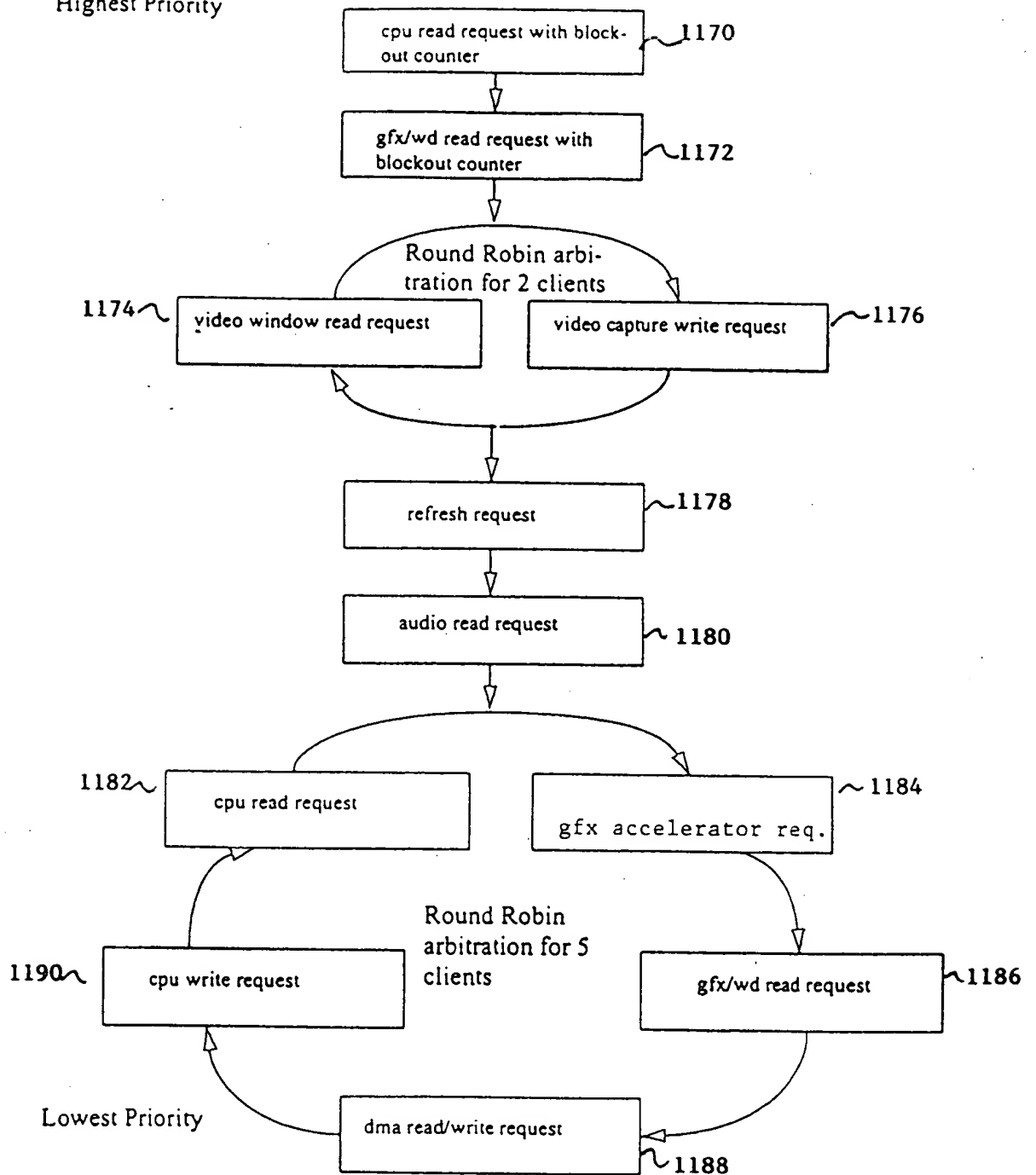


FIG. 36

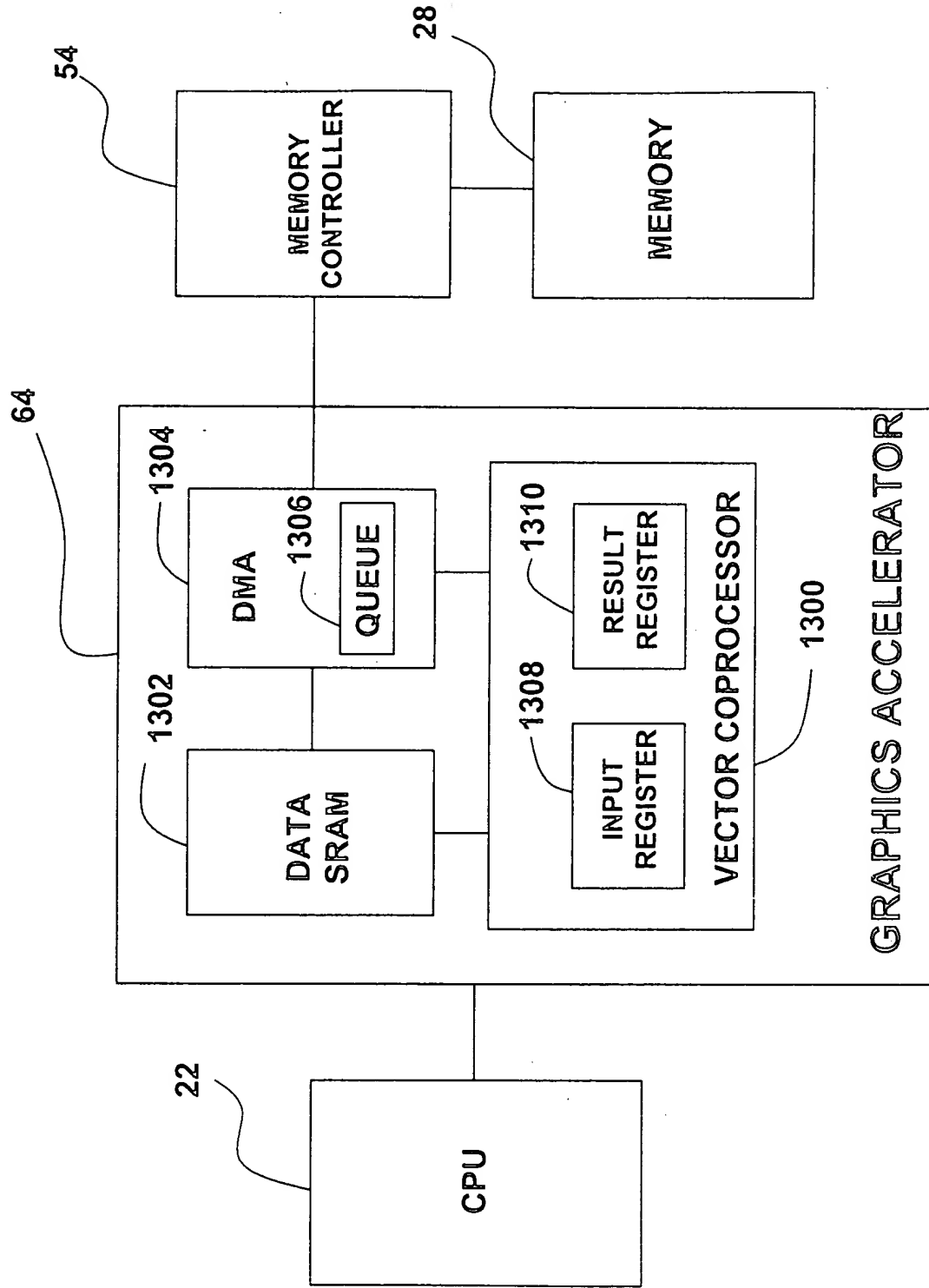


FIG. 37

09437590 100999